```
                  CRAXCLADTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
            BOTSECT=1
            DO 180 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
               IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(C)).LT.
     c         SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT))) THEN
                  BOTSECT=C
               ENDIF
  180       CONTINUE
*        Define the CR cladding bottom surface.
               IF (SECT.EQ.BOTSECT) THEN
                 CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c           CRADIM(BANKNUM(COLUMN,ROW),6)
               ELSE
                 CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               ENDIF
                 IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                    CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 190 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                       ENDIF
                    ENDIF
  190             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     CRAXCLADBOTTOMSURF(SECT)=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     CRAXCLADBOTTOMSURF(SECT)=CURRENTSURFLABEL
                  ENDIF
  200 CONTINUE
      DO 250 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
                  CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
                  IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                     CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                  ENDIF
                  CURRENTSURFLABEL=0
                  DO 210 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                            CURRENTSURFLABEL=V
                            EXIT
                        ENDIF
                     ENDIF
  210             CONTINUE
```

```
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTTOPSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section bottom surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
                CURRENTSURFLABEL=0
                DO 220 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 220            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTBOTSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section outer radius surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
                CURRENTSURFLABEL=0
                DO 230 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 230            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTORSURF(SECT)=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTORSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section inner radius surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
                CURRENTSURFLABEL=0
                DO 240 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
```

```
                  ENDIF
               ENDIF
  240        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
               GTSECTIRSURF(SECT)=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
             ELSE
               GTSECTIRSURF(SECT)=CURRENTSURFLABEL
             ENDIF
  250 CONTINUE
      DO 670 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
        IF (SURFVALUESPEC(CRAXABSBOTTOMSURF(SECT)).LT.
     c  SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
             CRABSML=MN
             DO 280 C=1,CRABSAXMAT(BANKNUM(COLUMN,ROW),2,SECT)
               IF (C.EQ.1) THEN
                 WRITE(200,260) CRABSML,
     c           CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c           (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT)),
     c           ASSYID(COLUMN,ROW)
  260            FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c           '   $ Control Rod Absorber Material in Assembly ',
     c           A5)
               ELSE
                 WRITE(200,270)
     c           CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c           (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT))
  270            FORMAT(T9,A9,3X,G14.6)
               ENDIF
  280        CONTINUE
             MN=MN+1
*     Write the CR absorber cell in this CR universe.
           WRITE(30,290) LN, CRABSML,
     c     (-1*CRABSAXMAT(BANKNUM(COLUMN,ROW),1,SECT)),
     c     (-1*CRAXABSSURF(SECT)),
     c     (-1*CRAXABSTOPSURF(SECT)), CRAXABSBOTTOMSURF(SECT),
     c     CRAUNIV(COLUMN,ROW)
  290      FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ Control rod absorber material')
           LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
           WRITE(30,300) LN, (-1*CRAXCLADIRSURF(SECT)),
     c     CRAXABSSURF(SECT), (-1*CRAXABSTOPSURF(SECT)),
     c     CRAXABSBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
  300      FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ Absorber-to-cladding gap')
           LN=LN+1
        ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
```

```
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 320 RO=1,(ROW-1)
                  DO 310 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                        IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=
     c                     CRAXCLADML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
  310             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  320          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 340 RO=ROW,ROW
                     DO 330 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                        IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                  CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRAXCLADML(COLUMN,ROW,SECT)=
     c                     CRAXCLADML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
  330                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
```

```
                        EXIT
                      ENDIF
 340                CONTINUE
                   ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 360 RO=1,(ROW-1)
                    DO 350 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                          IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                    CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                             EXIT
                          ENDIF
                          IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                    CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=
     c                       CRAXCLADML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
                       ENDIF
 350                CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 360             CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 380 RO=1,1
                    DO 370 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                          IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                    CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                             EXIT
                          ENDIF
                          IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                    CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=
     c                       CRAXCLADML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
                       ENDIF
 370                CONTINUE
```

```
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  380           CONTINUE
              ENDIF
           IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(SECT)).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
              IF (CLADMLUNIQUE.EQ..TRUE.) THEN
              CRAXCLADML(COLUMN,ROW,SECT)=MN
*   Check Control Rod Cladding
                  IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 390 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) CRAXCLADML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
  390                CONTINUE
                  ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c            .EQ.2) THEN
                     DO 400 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) CRAXCLADML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
                           WRITE(200,9311)
                           WRITE(200,9312)
                           WRITE(200,7009)
                           WRITE(200,7010)
                           WRITE(200,7011)
                           WRITE(200,9313)
                           WRITE(200,7012)
                           WRITE(200,7013)
                           WRITE(200,7014)
                           WRITE(200,7015)
                        ENDIF
  400                CONTINUE
                  ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
```

```
     c             .EQ.3) THEN
                      DO 410 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9314) CRAXCLADML(COLUMN,ROW,SECT)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9315)
                            WRITE(200,9316)
                            WRITE(200,9317)
                            WRITE(200,9318)
                            WRITE(200,7016)
                            WRITE(200,7017)
                            WRITE(200,7018)
                            WRITE(200,9319)
                            WRITE(200,9320)
                            WRITE(200,7019)
                            WRITE(200,7020)
                            WRITE(200,7021)
                            WRITE(200,9321)
                            WRITE(200,7022)
                            WRITE(200,7023)
                            WRITE(200,7024)
                            WRITE(200,7025)
                            WRITE(200,9322)
                            WRITE(200,9323)
                            WRITE(200,9324)
                            WRITE(200,9325)
                            WRITE(200,9326)
                            WRITE(200,9327)
                            WRITE(200,7026)
                            WRITE(200,9328)
                            WRITE(200,9329)
                            WRITE(200,9330)
                         ENDIF
   410                CONTINUE
                    ENDIF
                 MN=MN+1
              ENDIF
              IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                 CLADRHO=6.56
              ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.2) THEN
                 CLADRHO=7.90
              ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.3) THEN
                 CLADRHO=8.19
              ENDIF
              WRITE(30,420) LN, CRAXCLADML(COLUMN,ROW,SECT),
     c        (-1*CLADRHO), CRAXCLADIRSURF(SECT),
     c        (-1*CRAXCLADORSURF(SECT)), (-1*CRAXCLADTOPSURF(SECT)),
     c        CRAXCLADBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
   420        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod cladding')
              LN=LN+1
           ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
```

```
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 440 RO=1,(ROW-1)
                  DO 430 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
  430             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  440          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 460 RO=ROW,ROW
                     DO 450 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
  450                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  460             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 480 RO=1,(ROW-1)
                  DO 470 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
  470             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
```

```
                  ENDIF
  480           CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 500 RO=1,1
                    DO 490 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             CRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
  490                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  500             CONTINUE
                ENDIF
           IF (SURFVALUESPEC(CRAXABSTOPSURF(TOPSECT)).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
              IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
                 CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
                 DO 530 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                       WRITE(200,510) CRUPML(COLUMN,ROW),
     c                 CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
  510                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '   $ Control Rod Upper Plenum')
                    ELSE
                       WRITE(200,520)
     c                 CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
  520                  FORMAT(T9,A9,3X,G14.6)
                    ENDIF
  530               CONTINUE
                 MN=MN+1
              ENDIF
              WRITE(30,540) LN, CRUPML(COLUMN,ROW),
     c        (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c        CRAXABSTOPSURF(TOPSECT),
     c        (-1*CRAXCLADTOPSURF(TOPSECT)),
     c        (-1*CRAXCLADIRSURF(TOPSECT)),
     c        CRAUNIV(COLUMN,ROW)
  540         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
              LN=LN+1
           ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
```

```
*     the lower plenum material specification label.
            CRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 560 RO=1,(ROW-1)
                  DO 550 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                           CRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 550              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 560           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 580 RO=ROW,ROW
                     DO 570 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              CRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 570                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 580              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 600 RO=1,(ROW-1)
                  DO 590 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 590              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 600           CONTINUE
```

```
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 620 RO=1,1
                  DO 610 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
  610             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  620          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT)).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
               CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
               DO 650 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,630) CRLPML(COLUMN,ROW),
     c               CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
  630                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '   $ Control Rod Lower Plenum')
                  ELSE
                     WRITE(200,640)
     c               CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
  640                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
  650          CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,660) LN, CRLPML(COLUMN,ROW),
     c      (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      CRAXCLADBOTTOMSURF(BOTSECT),
     c      (-1*CRAXABSBOTTOMSURF(BOTSECT)),
     c      (-1*CRAXCLADIRSURF(BOTSECT)),
     c      CRAUNIV(COLUMN,ROW)
  660       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod lower plenum')
            LN=LN+1
         ENDIF
  670    CONTINUE
*     Write the GT material cell
         DO 800 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
```

```
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 690 RO=1,(ROW-1)
                  DO 680 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
  680             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  690          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 710 RO=ROW,ROW
                     DO 700 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
  700                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  710             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 730 RO=1,(ROW-1)
```

```
                  DO 720 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
  720             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  730          CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 750 RO=1,1
                     DO 740 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                     (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
  740                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  750             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                  IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 760 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
```

```
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9301)
                     WRITE(200,7000)
                     WRITE(200,7001)
                     WRITE(200,7002)
                     WRITE(200,9302)
                     WRITE(200,7003)
                     WRITE(200,7004)
                     WRITE(200,7005)
                     WRITE(200,9303)
                     WRITE(200,9304)
                  ENDIF
760            CONTINUE
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c        .EQ.2) THEN
                 DO 770 C=1,2
                    IF (C.EQ.1) THEN
                       WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                    ELSEIF (C.EQ.2) THEN
                       WRITE(200,9306)
                       WRITE(200,9307)
                       WRITE(200,9308)
                       WRITE(200,9309)
                       WRITE(200,9310)
                       WRITE(200,7006)
                       WRITE(200,7007)
                       WRITE(200,7008)
                       WRITE(200,9311)
                       WRITE(200,9312)
                       WRITE(200,7009)
                       WRITE(200,7010)
                       WRITE(200,7011)
                       WRITE(200,9313)
                       WRITE(200,7012)
                       WRITE(200,7013)
                       WRITE(200,7014)
                       WRITE(200,7015)
                    ENDIF
770              CONTINUE
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c        .EQ.3) THEN
                 DO 780 C=1,2
                    IF (C.EQ.1) THEN
                       WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                    ELSEIF (C.EQ.2) THEN
                       WRITE(200,9315)
                       WRITE(200,9316)
                       WRITE(200,9317)
                       WRITE(200,9318)
                       WRITE(200,7016)
                       WRITE(200,7017)
                       WRITE(200,7018)
                       WRITE(200,9319)
                       WRITE(200,9320)
```

```
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
                        WRITE(200,9329)
                        WRITE(200,9330)
                      ENDIF
 780                CONTINUE
                  ENDIF
               MN=MN+1
            ENDIF
            IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
               CLADRHO=6.56
            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
               CLADRHO=7.90
            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,790) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c      GTSECTIRSURF(SECT),
     c      (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c      GTSECTBOTSURF(SECT), CRAUNIV(COLUMN,ROW)
 790        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Guide tube')
            LN=LN+1
 800     CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
        SPACHEIGHT=0.0
        DO 810 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
           SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 810     CONTINUE
        DO 1000 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
          IF (SPN.EQ.1) THEN
             SPACERTOPSURF=UEFBOTTOMSURF
             CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c       SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
             CURRENTSURFLABEL=0
             DO 820 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 820           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 830 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 830           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     C      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     C         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 840 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 840           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
```

```
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 850 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 850            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   WATERREGIONBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
             ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
                SPACERTOPSURF=WATERREGIONBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 860 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 860            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
          DO 990 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
             IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,870) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c       SPN
```

```
 870      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,880) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
 880      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,890) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
 890      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c    SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
          WRITE(30,900) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c    SPN
 900      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c    ' IMP:N=1 U=',I4,
     c    '   $ Homogenized region for spacer ',I2)
          LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c    SURFVALUESPEC(SPACERTOPSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c    SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c    SURFVALUESPEC(SPACERTOPSURF))) THEN
          WRITE(30,910) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c    (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c    GTSECTORSURF(SECT),
     c    (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c    CRAUNIV(COLUMN,ROW), SPN
```

```
  910       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,920) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW), SPN
  920       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
             IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,930) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
  930       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,940) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
  940       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,950) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
  950       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
```

```
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,960) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
  960       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
            WRITE(30,970) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c      CRAUNIV(COLUMN,ROW)
  970       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,980) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
  980       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ENDIF
  990       CONTINUE
 1000     CONTINUE
*  Write the moderator inside of the GT in the CR universe
       DO 1060 CRSECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
-         DO 1050 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
            IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c      SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c      SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
*     Write the moderator cells within the GT in this CR universe.
               WRITE(30,1010) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c         CRAXCLADBOTTOMSURF(CRSECT),
     c         CRAUNIV(COLUMN,ROW)
```

```
 1010          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
               WRITE(30,1020) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c         GTSECTBOTSURF(GTSECT),
     c         CRAUNIV(COLUMN,ROW)
 1020          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
               WRITE(30,1030) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRAXCLADORSURF(CRSECT), (-1*GTSECTTOPSURF(GTSECT)),
     c         CRAXCLADBOTTOMSURF(CRSECT),
     c         CRAUNIV(COLUMN,ROW)
 1030          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
               WRITE(30,1040) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRAXCLADORSURF(CRSECT), (-1*GTSECTTOPSURF(GTSECT)),
     c         GTSECTBOTSURF(GTSECT),
     c         CRAUNIV(COLUMN,ROW)
 1040          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ENDIF
 1050      CONTINUE
 1060  CONTINUE
*    Determine the axial GT section which contains the lowest CR axial section
       DO 1070 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c    SURFVALUESPEC(CRAXCLADBOTTOMSURF
```

```
     c    (NUMCRAXS(BANKNUM(COLUMN,ROW)))).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c    SURFVALUESPEC(CRAXCLADBOTTOMSURF
     c    (NUMCRAXS(BANKNUM(COLUMN,ROW)))))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 1070 CONTINUE
      DO 1100 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,1080) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*CRAXCLADBOTTOMSURF(NUMCRAXS(BANKNUM(COLUMN,ROW)))),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 1080        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,1090) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 1090        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 1100 CONTINUE
*     Write the lower end-fitting cell specification for this CR universe.
          GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
          IF (SURFVALUESPEC(GTBOTSURF).GE.
     c    ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
             WRITE(30,1110) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c       CRAUNIV(COLUMN,ROW)
 1110        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c       '   $ Lower end-fitting')
             LN=LN+1
          ELSE
             WRITE(30,1120) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c       GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c       CRAUNIV(COLUMN,ROW)
 1120        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
             LN=LN+1
             WRITE(30,1130) LN, FRLEFML(COLUMN,ROW),
     c       (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c       (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c       (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c       CRAUNIV(COLUMN,ROW)
```

```
 1130          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
             ENDIF
*     Write the upper end-fitting cell specification for this CR universe.
             GTTOPSURF=GTSECTTOPSURF(1)
             GTSECTORSURF(1)=GTSECTORSURF(1)
             GTSECTIRSURF(1)=GTSECTIRSURF(1)
             CRCLADTOPSURF=CRAXCLADTOPSURF(1)
             CRCLADIRSURF=CRAXCLADIRSURF(1)
             CRCLADORSURF=CRAXCLADORSURF(1)
             IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1131) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1131           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1132) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1132           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1133) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1133           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1134) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 1134           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
```

```
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1135) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1135           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1136) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1136           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1137) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1137           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1138) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1138           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1139) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 1139           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1140           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
```

```
                LN=LN+1
                WRITE(30,1141) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1141           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c         SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c         (SURFVALUESPEC(GTTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1142) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1142           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1143) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1143           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1144) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*CRCLADTOPSURF), CRCLADORSURF,
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 1144           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1145) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 1145           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1146) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c          CRAUNIV(COLUMN,ROW)
 1146           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
```

```
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1147) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1147           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1148) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1148           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1149) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 1149           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1150) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 1150           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,1151) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
 1151           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1152) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
 1152           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
```

```
            ENDIF
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*
*
*
*
*
*
*
*    SPECIFICATIONS FOR HYBRID=1 AND GTSPLIT<>1
*
*
*
*
*
*
*
*
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
*******************************************************************************
```

```
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
      ELSEIF ((HYBRID.EQ.1).AND.(GTSPLIT.NE.1)) THEN
      DO 1280 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
*     Define the CR absorber radius.
               CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 1190 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1190          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRAXABSSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRAXABSSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c         CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1200 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1200          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRAXABSTOPSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRAXABSTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber bottom surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
```

```
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 1210 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
1210            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRAXABSBOTTOMSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   CRAXABSBOTTOMSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding inner radius.
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),2,SECT)
                CURRENTSURFLABEL=0
                DO 1220 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
1220            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRAXCLADIRSURF(SECT)=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   CRAXCLADIRSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding outer radius.
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),3,SECT)
                CURRENTSURFLABEL=0
                DO 1230 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
1230            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRAXCLADORSURF(SECT)=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
```

```
                ELSE
                   CRAXCLADORSURF(SECT)=CURRENTSURFLABEL
                ENDIF
              TOPSECT=1
              DO 1240 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
                 IF (SURFVALUESPEC(CRAXCLADTOPSURF(C)).GT.
     c           SURFVALUESPEC(CRAXCLADTOPSURF(TOPSECT))) THEN
                    TOPSECT=C
                 ENDIF
 1240         CONTINUE
*     Define the CR cladding top surface.
              IF (SECT.EQ.TOPSECT) THEN
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c          CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          CRADIM(BANKNUM(COLUMN,ROW),7)
              ELSE
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c          CRAXDIM(BANKNUM(COLUMN,ROW),5,SECT)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              ENDIF
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 1250 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1250            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    CRAXCLADTOPSURF(SECT)=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    CRAXCLADTOPSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
           BOTSECT=1
           DO 1260 C=2,NUMCRAXS(BANKNUM(COLUMN,ROW))
              IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(C)).LT.
      c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT))) THEN
                 BOTSECT=C
              ENDIF
 1260      CONTINUE
*     Define the CR cladding bottom surface.
              IF (SECT.EQ.BOTSECT) THEN
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
      c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
      c         CRADIM(BANKNUM(COLUMN,ROW),6)
              ELSE
```

```
                CURRENTSURF=CRAXDIM(BANKNUM(COLUMN,ROW),4,SECT)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              ENDIF
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 1270 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1270            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    CRAXCLADBOTTOMSURF(SECT)=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    CRAXCLADBOTTOMSURF(SECT)=CURRENTSURFLABEL
                 ENDIF
 1280 CONTINUE
*     Define the GT top surface.
                 CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
                 IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                    CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                 ENDIF
                 CURRENTSURFLABEL=0
                 DO 1290 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 1290            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    GTTOPSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    GTTOPSURF=CURRENTSURFLABEL
                 ENDIF
*     Define the GT bottom surface.
                 CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
                 CURRENTSURFLABEL=0
                 DO 1300 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
```

```
                  ENDIF
               ENDIF
 1300          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTBOTSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTBOTSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT outer radius surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1310 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1310          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT inner radius surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 1320 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1320          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTIRSURF=CURRENTSURFLABEL
               ENDIF
*     Write the GT material cell in this GT universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
         CLADMLUNIQUE=.TRUE.
```

```
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1340 RO=1,(ROW-1)
                  DO 1330 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1330             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1340          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1360 RO=ROW,ROW
                     DO 1350 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1350                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1360             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 1380 RO=1,(ROW-1)
                  DO 1370 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1370             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
```

```
                  ENDIF
 1380           CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 1400 RO=1,1
                    DO 1390 CO=1,(COLUMN-1)
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTML(COLUMN,ROW)=GTML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 1390                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1400             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
                  IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 1410 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
 1410                CONTINUE
                    ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c              .EQ.2) THEN
                     DO 1420 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
```

```
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
1420                 CONTINUE
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c            .EQ.3) THEN
                     DO 1430 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
                           WRITE(200,9317)
                           WRITE(200,9318)
                           WRITE(200,7016)
                           WRITE(200,7017)
                           WRITE(200,7018)
                           WRITE(200,9319)
                           WRITE(200,9320)
                           WRITE(200,7019)
                           WRITE(200,7020)
                           WRITE(200,7021)
                           WRITE(200,9321)
                           WRITE(200,7022)
                           WRITE(200,7023)
                           WRITE(200,7024)
                           WRITE(200,7025)
                           WRITE(200,9322)
                           WRITE(200,9323)
                           WRITE(200,9324)
                           WRITE(200,9325)
                           WRITE(200,9326)
                           WRITE(200,9327)
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
1430                 CONTINUE
                  ENDIF
               MN=MN+1
            ENDIF
            IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
               CLADRHO=6.56
            ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
```

```
                CLADRHO=7.90
              ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                CLADRHO=8.19
              ENDIF
              WRITE(30,1440) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c        GTIRSURF,
     c        (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c        GTUNIV(COLUMN,ROW)
 1440         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Guide tube')
              LN=LN+1
               SPACHEIGHT=0.0
*      Loop through the spacer and moderator regions along the axial
*      length of the GT (from top to bottom).
            DO 1450 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 1450       CONTINUE
            DO 1530 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*      Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                SPACERTOPSURF=UEFBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 1460 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1460           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 1470 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1470           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   WATERREGIONBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
```

```
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
              ELSE
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
              SPACERTOPSURF=WATERREGIONBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 1480 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
1480          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                SPACERBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
              ELSE
                SPACERBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
              WATERREGIONTOPSURF=SPACERBOTTOMSURF
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
              CURRENTSURFLABEL=0
              DO 1490 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
1490          CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                WATERREGIONBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
              ELSE
                WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
              ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
              SPACERTOPSURF=WATERREGIONBOTTOMSURF
              CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c        SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
              CURRENTSURFLABEL=0
              DO 1500 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1500          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
            WRITE(30,1510) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, GTUNIV(COLUMN,ROW),
     c      SPN
 1510       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
            WRITE(30,1520) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      GTUNIV(COLUMN,ROW)
 1520       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
 1530    CONTINUE
      DO 1950 SECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
         IF (SURFVALUESPEC(CRAXABSBOTTOMSURF(SECT)).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
               CRABSML=MN
               DO 1560 C=1,CRABSAXMAT(BANKNUM(COLUMN,ROW),2,SECT)
                  IF (C.EQ.1) THEN
                     WRITE(200,1540) CRABSML,
     c               CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c               (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT)),
     c               ASSYID(COLUMN,ROW)
 1540                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '   $ Control Rod Absorber Material in Assembly ',
     c               A5)
                  ELSE
                     WRITE(200,1550)
     c               CRABSAXZAIDS(BANKNUM(COLUMN,ROW),C,SECT),
     c               (-1*CRABSAXWTS(BANKNUM(COLUMN,ROW),C,SECT))
 1550                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
```

```
 1560          CONTINUE
               MN=MN+1
*     Write the CR absorber cell in this CR universe.
              WRITE(30,1570) LN, CRABSML,
     c        (-1*CRABSAXMAT(BANKNUM(COLUMN,ROW),1,SECT)),
     c        (-1*CRAXABSSURF(SECT)),
     c        (-1*CRAXABSTOPSURF(SECT)), CRAXABSBOTTOMSURF(SECT),
     c        CRAUNIV(COLUMN,ROW)
 1570         FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod absorber material')
              LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
              WRITE(30,1580) LN, (-1*CRAXCLADIRSURF(SECT)),
     c        CRAXABSSURF(SECT), (-1*CRAXABSTOPSURF(SECT)),
     c        CRAXABSBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
 1580         FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Absorber-to-cladding gap')
              LN=LN+1
           ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
              CLADMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 1600 RO=1,(ROW-1)
                    DO 1590 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                          IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                    CRCLADMAT(BANKNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                             EXIT
                          ENDIF
                          IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                    CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRAXCLADML(COLUMN,ROW,SECT)=
     c                       CRAXCLADML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
                       ENDIF
 1590               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 1600            CONTINUE
                 IF (LEAVE.EQ..FALSE.) THEN
                    DO 1620 RO=ROW,ROW
```

```
                  DO 1610 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                     IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                        EXIT
                     ENDIF
                     IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=
     c                  CRAXCLADML(CO,RO,SECT)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
1610              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
1620           CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 1640 RO=1,(ROW-1)
               DO 1630 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                     IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                        EXIT
                     ENDIF
                     IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c               CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRAXCLADML(COLUMN,ROW,SECT)=
     c                  CRAXCLADML(CO,RO,SECT)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
1630           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
1640        CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
```

```
                DO 1660 RO=1,1
                   DO 1650 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                         IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRAXCLADML(COLUMN,ROW,SECT)=CRCLADML(CO,RO)
                            EXIT
                         ENDIF
                         IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.
     c                   CRAXCLADMAT(BANKNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRAXCLADML(COLUMN,ROW,SECT)=
     c                      CRAXCLADML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 1650              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 1660           CONTINUE
             ENDIF
          IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(SECT)).LT.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                CRAXCLADML(COLUMN,ROW,SECT)=MN
*  Check Control Rod Cladding
                IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                   DO 1670 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9300) CRAXCLADML(COLUMN,ROW,SECT)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9301)
                         WRITE(200,7000)
                         WRITE(200,7001)
                         WRITE(200,7002)
                         WRITE(200,9302)
                         WRITE(200,7003)
                         WRITE(200,7004)
                         WRITE(200,7005)
                         WRITE(200,9303)
                         WRITE(200,9304)
                      ENDIF
 1670              CONTINUE
                ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     c          .EQ.2) THEN
                   DO 1680 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9305) CRAXCLADML(COLUMN,ROW,SECT)
```

```
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
                           WRITE(200,9311)
                           WRITE(200,9312)
                           WRITE(200,7009)
                           WRITE(200,7010)
                           WRITE(200,7011)
                           WRITE(200,9313)
                           WRITE(200,7012)
                           WRITE(200,7013)
                           WRITE(200,7014)
                           WRITE(200,7015)
                        ENDIF
1680                    CONTINUE
                     ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT)
     C               .EQ.3) THEN
                        DO 1690 C=1,2
                           IF (C.EQ.1) THEN
                              WRITE(200,9314) CRAXCLADML(COLUMN,ROW,SECT)
                           ELSEIF (C.EQ.2) THEN
                              WRITE(200,9315)
                              WRITE(200,9316)
                              WRITE(200,9317)
                              WRITE(200,9318)
                              WRITE(200,7016)
                              WRITE(200,7017)
                              WRITE(200,7018)
                              WRITE(200,9319)
                              WRITE(200,9320)
                              WRITE(200,7019)
                              WRITE(200,7020)
                              WRITE(200,7021)
                              WRITE(200,9321)
                              WRITE(200,7022)
                              WRITE(200,7023)
                              WRITE(200,7024)
                              WRITE(200,7025)
                              WRITE(200,9322)
                              WRITE(200,9323)
                              WRITE(200,9324)
                              WRITE(200,9325)
                              WRITE(200,9326)
                              WRITE(200,9327)
                              WRITE(200,7026)
                              WRITE(200,9328)
                              WRITE(200,9329)
                              WRITE(200,9330)
```

```
                        ENDIF
 1690                 CONTINUE
                     ENDIF
                  MN=MN+1
               ENDIF
               IF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.1) THEN
                  CLADRHO=6.56
               ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.2) THEN
                  CLADRHO=7.90
               ELSEIF (CRAXCLADMAT(BANKNUM(COLUMN,ROW),SECT).EQ.3) THEN
                  CLADRHO=8.19
               ENDIF
               WRITE(30,1700) LN, CRAXCLADML(COLUMN,ROW,SECT),
     c         (-1*CLADRHO), CRAXCLADIRSURF(SECT),
     c         (-1*CRAXCLADORSURF(SECT)), (-1*CRAXCLADTOPSURF(SECT)),
     c         CRAXCLADBOTTOMSURF(SECT), CRAUNIV(COLUMN,ROW)
 1700        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Control rod cladding')
               LN=LN+1
            ENDIF
*        Write the CR upper plenum cell in this CR universe.
*        Determine if the CR upper plenum material specification has
*        previously been defined.  If it has been previously defined, determine
*        the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1720 RO=1,(ROW-1)
                  DO 1710 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1710             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1720          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1740 RO=ROW,ROW
                     DO 1730 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                              EXIT
                           ENDIF
```

```
                        ENDIF
 1730                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 1740                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 1760 RO=1,(ROW-1)
                     DO 1750 CO=1,50
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1750                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1760             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 1780 RO=1,1
                     DO 1770 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1770                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1780             CONTINUE
               ENDIF
            IF (SURFVALUESPEC(CRAXABSTOPSURF(TOPSECT)).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
               IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
                  CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
                  DO 1810 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,1790) CRUPML(COLUMN,ROW),
     c                  CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1790                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '   $ Control Rod Upper Plenum')
```

```
                  ELSE
                     WRITE(200,1800)
     c               CRUPZS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1800                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 1810             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,1820) LN, CRUPML(COLUMN,ROW),
     c      (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      CRAXABSTOPSURF(TOPSECT),
     c      (-1*CRAXCLADTOPSURF(TOPSECT)),
     c      (-1*CRAXCLADIRSURF(TOPSECT)),
     c      CRAUNIV(COLUMN,ROW)
 1820       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
            LN=LN+1
         ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            CRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1840 RO=1,(ROW-1)
                  DO 1830 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                           CRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1830             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1840          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1860 RO=ROW,ROW
                     DO 1850 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              CRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1850                CONTINUE
```

```
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
1860           CONTINUE
            ENDIF
         ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
            DO 1880 RO=1,(ROW-1)
               DO 1870 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
1870           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
1880        CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 1900 RO=1,1
               DO 1890 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        CRLPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
1890           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
1900        CONTINUE
         ENDIF
      IF (SURFVALUESPEC(CRAXCLADBOTTOMSURF(BOTSECT)).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
            CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
            DO 1930 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
               IF (C.EQ.1) THEN
                  WRITE(200,1910) CRLPML(COLUMN,ROW),
     c            CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c            (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
1910              FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c            '   $ Control Rod Lower Plenum')
               ELSE
                  WRITE(200,1920)
```

```
     c                    CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                    (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1920                     FORMAT(T9,A9,3X,G14.6)
                        ENDIF
 1930                CONTINUE
                   MN=MN+1
                ENDIF
                WRITE(30,1940) LN, CRLPML(COLUMN,ROW),
     c          (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c          CRAXCLADBOTTOMSURF(BOTSECT),
     c          (-1*CRAXABSBOTTOMSURF(BOTSECT)),
     c          (-1*CRAXCLADIRSURF(BOTSECT)),
     c          CRAUNIV(COLUMN,ROW)
 1940          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Control rod lower plenum')
               LN=LN+1
            ENDIF
 1950       CONTINUE
*  Write the moderator inside of the GT in the CR universe
         DO 2000 CRSECT=1,NUMCRAXS(BANKNUM(COLUMN,ROW))
               IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c         SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c         (SURFVALUESPEC(GTBOTSURF).LE.
     c         SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
*     Write the moderator cells within the GT in this CR universe.
                  WRITE(30,1960) LN, BMODML, (-1*MODDENSITY),
     c            (-1*GTIRSURF),
     c            CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c            CRAXCLADBOTTOMSURF(CRSECT),
     c            CRAUNIV(COLUMN,ROW)
 1960             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,
     c            '   $ Borated moderator inside guide tube')
                  LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c         SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c         (SURFVALUESPEC(GTBOTSURF).LT.
     c         SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c         (SURFVALUESPEC(GTBOTSURF).GT.
     c         SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                  WRITE(30,1970) LN, BMODML, (-1*MODDENSITY),
     c            (-1*GTIRSURF),
     c            CRAXCLADORSURF(CRSECT), (-1*CRAXCLADTOPSURF(CRSECT)),
     c            GTBOTSURF,
     c            CRAUNIV(COLUMN,ROW)
 1970             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,
     c            '   $ Borated moderator inside guide tube')
                  LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c         SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c         (SURFVALUESPEC(GTBOTSURF).LE.
     c         SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT))).AND.
     c         (SURFVALUESPEC(GTTOPSURF).GT.
```

```
     c       SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT))) THEN
                WRITE(30,1980) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTIRSURF),
     c          CRAXCLADORSURF(CRSECT), (-1*GTTOPSURF),
     c          CRAXCLADBOTTOMSURF(CRSECT),
     c          CRAUNIV(COLUMN,ROW)
 1980           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(CRAXCLADTOPSURF(CRSECT))).AND.
     c        (SURFVALUESPEC(GTBOTSURF).GT.
     c        SURFVALUESPEC(CRAXCLADBOTTOMSURF(CRSECT)))) THEN
                WRITE(30,1990) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTIRSURF),
     c          CRAXCLADORSURF(CRSECT), (-1*GTTOPSURF),
     c          GTBOTSURF,
     c          CRAUNIV(COLUMN,ROW)
 1990           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ENDIF
 2000 CONTINUE
              WRITE(30,2010) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTIRSURF),
     c        (-1*CRAXCLADBOTTOMSURF(NUMCRAXS(BANKNUM(COLUMN,ROW)))),
     c        GTBOTSURF,
     c        CRAUNIV(COLUMN,ROW)
 2010         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '   $ Borated moderator inside guide tube')
              LN=LN+1
*      Write the lower end-fitting cell specification for this CR universe.
              IF (SURFVALUESPEC(GTBOTSURF).GE.
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
                WRITE(30,2020) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c          CRAUNIV(COLUMN,ROW)
 2020           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c          '   $ Lower end-fitting')
                LN=LN+1
              ELSE
                WRITE(30,2030) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c          GTORSURF,
     c          CRAUNIV(COLUMN,ROW)
 2030           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
                LN=LN+1
                WRITE(30,2040) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c          (-1*GTBOTSURF),
```

```
     c          (-1*GTORSURF),
     c          CRAUNIV(COLUMN,ROW)
 2040           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
                LN=LN+1
              ENDIF
*     Write the upper end-fitting cell specification for this CR universe.
              GTTOPSURF=GTTOPSURF
              GTSECTORSURF(1)=GTORSURF
              GTSECTIRSURF(1)=GTIRSURF
              CRCLADTOPSURF=CRAXCLADTOPSURF(1)
              CRCLADIRSURF=CRAXCLADIRSURF(1)
              CRCLADORSURF=CRAXCLADORSURF(1)
              IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,2050) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2050            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,2051) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2051            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,2052) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c           (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c           CRAUNIV(COLUMN,ROW)
 2052            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,2053) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 2053            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
```

```
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2054) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2054           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,2055) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 2055           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2056) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2056           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,2057) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 2057           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,2058) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 2058           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,2059) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
```

```
 2059          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
               WRITE(30,2060) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c         CRAUNIV(COLUMN,ROW)
 2060          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,2061) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 2061          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
               WRITE(30,2062) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c         CRAUNIV(COLUMN,ROW)
 2062          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
               WRITE(30,2063) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c         (-1*CRCLADTOPSURF), CRCLADORSURF,
     c         (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
 2063          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,2064) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 2064          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
               WRITE(30,2065) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c         CRAUNIV(COLUMN,ROW)
```

```
2065          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
             ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GE.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,2066) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
2066          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2067) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c        CRAUNIV(COLUMN,ROW)
2067          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
             ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,2068) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
2068          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2069) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c        CRAUNIV(COLUMN,ROW)
2069          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,2070) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
2070          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
             ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,2071) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
2071          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
```

```
      c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
               ENDIF
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*
*
*
*
*
*
*
*   SPECIFICATIONS FOR HYBRID<>1 AND GTSPLIT=1
*
*
*
*
*
*
*
*
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
*********************************************************************************
```

```
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
      ELSEIF ((HYBRID.NE.1).AND.(GTSPLIT.EQ.1)) THEN
*     Define the CR absorber radius.
               CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 2100 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2100          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRABSSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRABSSURF=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRADIM(BANKNUM(COLUMN,ROW),4)+
     c         CRADIM(BANKNUM(COLUMN,ROW),5)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 2110 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2110          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRABSTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  CRABSTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the CR absorber bottom surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c         CRADIM(BANKNUM(COLUMN,ROW),4)
```

```
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2120 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2120           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRABSBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   CRABSBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding inner radius.
                CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 2130 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2130           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRCLADIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   CRCLADIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding outer radius.
                CURRENTSURF=CRADIM(BANKNUM(COLUMN,ROW),3)
                CURRENTSURFLABEL=0
                DO 2140 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2140           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   CRCLADORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
```

```
                  SN=SN+1
                ELSE
                  CRCLADORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding top surface.
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          CRADIM(BANKNUM(COLUMN,ROW),4)+
     c          CRADIM(BANKNUM(COLUMN,ROW),5)+
     c          CRADIM(BANKNUM(COLUMN,ROW),7)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2150 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 2150           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRCLADTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  CRCLADTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the CR cladding bottom surface.
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)+
     c          CRADIM(BANKNUM(COLUMN,ROW),4)-
     c          CRADIM(BANKNUM(COLUMN,ROW),6)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2160 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 2160           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  CRCLADBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  CRCLADBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
          IF (SURFVALUESPEC(CRABSBOTTOMSURF).LT.
```

```
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*  Check Control Rod Absorber Material
               CRABSML=MN
               DO 2190 C=1,CRABSMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,2170) CRABSML,
     c               CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C)),
     c               ASSYID(COLUMN,ROW)
 2170                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c               '   $ Control Rod Absorber Material in Assembly ',
     c               A5)
                  ELSE
                     WRITE(200,2180) CRABSZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*CRABSWTS(BANKNUM(COLUMN,ROW),C))
 2180                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 2190          CONTINUE
               MN=MN+1
*     Write the CR absorber cell in this CR universe.
            WRITE(30,2200) LN, CRABSML,
     c      (-1*CRABSMAT(BANKNUM(COLUMN,ROW),1)), (-1*CRABSSURF),
     c      (-1*CRABSTOPSURF), CRABSBOTTOMSURF,
     c      CRAUNIV(COLUMN,ROW)
 2200       FORMAT(T1,I4,T6,I4,T11,F10.6,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod absorber material')
            LN=LN+1
*     Write the absorber-to-cladding gap cell in this CR universe.
            WRITE(30,2210) LN, (-1*CRCLADIRSURF), CRABSSURF,
     c      (-1*CRABSTOPSURF),
     c      CRABSBOTTOMSURF, CRAUNIV(COLUMN,ROW)
 2210       FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Absorber-to-cladding gap')
            LN=LN+1
         ENDIF
*     Write the CR cladding cell in this CR universe.
*     Determine if the CR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2230 RO=1,(ROW-1)
                  DO 2220 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
```

```
                  ENDIF
2220              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
2230           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2250 RO=ROW,ROW
                     DO 2240 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
2240                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
2250              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2270 RO=1,(ROW-1)
                  DO 2260 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
2260              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
2270           CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2290 RO=1,1
                  DO 2280 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'CRA  ') THEN
                        IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  CRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
```

```
                              CRCLADML(COLUMN,ROW)=CRCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 2280                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 2290              CONTINUE
                ENDIF
             IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)) THEN
                IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                   CRCLADML(COLUMN,ROW)=MN
*  Check Control Rod Cladding
                   IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                      DO 2300 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9300) CRCLADML(COLUMN,ROW)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9301)
                            WRITE(200,7000)
                            WRITE(200,7001)
                            WRITE(200,7002)
                            WRITE(200,9302)
                            WRITE(200,7003)
                            WRITE(200,7004)
                            WRITE(200,7005)
                            WRITE(200,9303)
                            WRITE(200,9304)
                         ENDIF
 2300                 CONTINUE
                    ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
     c              .EQ.2) THEN
                      DO 2310 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9305) CRCLADML(COLUMN,ROW)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9306)
                            WRITE(200,9307)
                            WRITE(200,9308)
                            WRITE(200,9309)
                            WRITE(200,9310)
                            WRITE(200,7006)
                            WRITE(200,7007)
                            WRITE(200,7008)
                            WRITE(200,9311)
                            WRITE(200,9312)
                            WRITE(200,7009)
                            WRITE(200,7010)
                            WRITE(200,7011)
                            WRITE(200,9313)
                            WRITE(200,7012)
```

```
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
2310                 CONTINUE
                   ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW))
     c             .EQ.3) THEN
                     DO 2320 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) CRCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
                           WRITE(200,9317)
                           WRITE(200,9318)
                           WRITE(200,7016)
                           WRITE(200,7017)
                           WRITE(200,7018)
                           WRITE(200,9319)
                           WRITE(200,9320)
                           WRITE(200,7019)
                           WRITE(200,7020)
                           WRITE(200,7021)
                           WRITE(200,9321)
                           WRITE(200,7022)
                           WRITE(200,7023)
                           WRITE(200,7024)
                           WRITE(200,7025)
                           WRITE(200,9322)
                           WRITE(200,9323)
                           WRITE(200,9324)
                           WRITE(200,9325)
                           WRITE(200,9326)
                           WRITE(200,9327)
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
2320                 CONTINUE
                   ENDIF
                MN=MN+1
             ENDIF
             IF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                CLADRHO=6.56
             ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                CLADRHO=7.90
             ELSEIF (CRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                CLADRHO=8.19
             ENDIF
             WRITE(30,2330) LN, CRCLADML(COLUMN,ROW), (-1*CLADRHO),
     c       CRCLADIRSURF,
     c       (-1*CRCLADORSURF), (-1*CRCLADTOPSURF), CRCLADBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
```

```
2330        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod cladding')
            LN=LN+1
          ENDIF
*     Write the CR upper plenum cell in this CR universe.
*     Determine if the CR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            CRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2350 RO=1,(ROW-1)
                  DO 2340 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
2340              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
2350           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 2370 RO=ROW,ROW
                     DO 2360 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              CRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
2360                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
2370              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2390 RO=1,(ROW-1)
                  DO 2380 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
```

```
                        EXIT
                      ENDIF
                    ENDIF
 2380             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 2390           CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 2410 RO=1,1
                    DO 2400 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             CRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             CRUPML(COLUMN,ROW)=CRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 2400                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                     ENDIF
 2410             CONTINUE
                ENDIF
           IF (SURFVALUESPEC(CRABSTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
              IF (CRUPMLUNIQUE.EQ..TRUE.) THEN
                 CRUPML(COLUMN,ROW)=MN
*  Check Control Rod Upper Plenum Regions
                 DO 2440 C=1,CRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                       WRITE(200,2420) CRUPML(COLUMN,ROW),
     c                 CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2420                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '   $ Control Rod Upper Plenum')
                    ELSE
                       WRITE(200,2430)
     c                 CRUPZS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*CRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2430                  FORMAT(T9,A9,3X,G14.6)
                    ENDIF
 2440             CONTINUE
                 MN=MN+1
              ENDIF
              WRITE(30,2450) LN, CRUPML(COLUMN,ROW),
     c        (-1*CRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c        CRABSTOPSURF,
     c        (-1*CRCLADTOPSURF), (-1*CRCLADIRSURF),
     c        CRAUNIV(COLUMN,ROW)
 2450         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Control rod upper plenum')
```

```
            LN=LN+1
          ENDIF
*     Write the CR lower plenum cell in this CR universe.
*     Determine if the CR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
          CRLPMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
             DO 2470 RO=1,(ROW-1)
                DO 2460 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                         CRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
2460            CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
2470         CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 2490 RO=ROW,ROW
                   DO 2480 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                            CRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
2480               CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
2490            CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 2510 RO=1,(ROW-1)
                DO 2500 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                BANKNUM(CO,RO)) THEN
                         CRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
```

```
 2500            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 2510          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2530 RO=1,1
                  DO 2520 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           CRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           CRLPML(COLUMN,ROW)=CRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 2520             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2530          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(CRCLADBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (CRLPMLUNIQUE.EQ..TRUE.) THEN
               CRLPML(COLUMN,ROW)=MN
*  Check Control Rod Lower Plenum Regions
                  DO 2560 C=1,CRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,2540) CRLPML(COLUMN,ROW),
     c                  CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 2540                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '   $ Control Rod Lower Plenum')
                     ELSE
                        WRITE(200,2550)
     c                  CRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*CRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 2550                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 2560             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,2570) LN, CRLPML(COLUMN,ROW),
     c      (-1*CRLPLENMAT(BANKNUM(COLUMN,ROW),1)), CRCLADBOTTOMSURF,
     c      (-1*CRABSBOTTOMSURF), (-1*CRCLADIRSURF),
     c      CRAUNIV(COLUMN,ROW)
 2570       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Control rod lower plenum')
            LN=LN+1
         ENDIF
      DO 2620 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
```

```
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
                IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2580 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2580           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTTOPSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section bottom surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
                CURRENTSURFLABEL=0
                DO 2590 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2590           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTBOTSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section outer radius surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
                CURRENTSURFLABEL=0
                DO 2600 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2600           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTORSURF(SECT)=SN
                   SURFTYPESPEC(SN)='CZ'
```

```
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 2610 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2610          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTIRSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTIRSURF(SECT)=CURRENTSURFLABEL
               ENDIF
 2620 CONTINUE
*     Write the GT material cell
          DO 2750 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 2640 RO=1,(ROW-1)
                  DO 2630 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 2630             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
```

```
                  ENDIF
 2640           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 2660 RO=ROW,ROW
                      DO 2650 CO=1,(COLUMN-1)
                         IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                            IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTMAT(DESNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                               EXIT
                            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                               EXIT
                            ENDIF
                         ENDIF
 2650                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 2660                 CONTINUE
                   ENDIF
                ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                   DO 2680 RO=1,(ROW-1)
                      DO 2670 CO=1,50
                         IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                            IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTMAT(DESNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                               EXIT
                            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                      GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                               EXIT
                            ENDIF
                         ENDIF
 2670                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 2680                 CONTINUE
                ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                   DO 2700 RO=1,1
                      DO 2690 CO=1,(COLUMN-1)
```

```
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                   (BANKNUM(CO,RO).EQ.0)) THEN
                         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                            EXIT
                         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF
 2690              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2700           CONTINUE
             ENDIF
             IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                   IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                      DO 2710 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9301)
                            WRITE(200,7000)
                            WRITE(200,7001)
                            WRITE(200,7002)
                            WRITE(200,9302)
                            WRITE(200,7003)
                            WRITE(200,7004)
                            WRITE(200,7005)
                            WRITE(200,9303)
                            WRITE(200,9304)
                         ENDIF
 2710                 CONTINUE
                   ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c             .EQ.2) THEN
                      DO 2720 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9306)
                            WRITE(200,9307)
                            WRITE(200,9308)
                            WRITE(200,9309)
                            WRITE(200,9310)
                            WRITE(200,7006)
```

```
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
2720              CONTINUE
                ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     C          .EQ.3) THEN
                   DO 2730 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9315)
                         WRITE(200,9316)
                         WRITE(200,9317)
                         WRITE(200,9318)
                         WRITE(200,7016)
                         WRITE(200,7017)
                         WRITE(200,7018)
                         WRITE(200,9319)
                         WRITE(200,9320)
                         WRITE(200,7019)
                         WRITE(200,7020)
                         WRITE(200,7021)
                         WRITE(200,9321)
                         WRITE(200,7022)
                         WRITE(200,7023)
                         WRITE(200,7024)
                         WRITE(200,7025)
                         WRITE(200,9322)
                         WRITE(200,9323)
                         WRITE(200,9324)
                         WRITE(200,9325)
                         WRITE(200,9326)
                         WRITE(200,9327)
                         WRITE(200,7026)
                         WRITE(200,9328)
                         WRITE(200,9329)
                         WRITE(200,9330)
                      ENDIF
2730               CONTINUE
                ENDIF
             MN=MN+1
          ENDIF
          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
             CLADRHO=6.56
```

```
            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
               CLADRHO=7.90
            ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,2740) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c      GTSECTIRSURF(SECT),
     c      (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c      GTSECTBOTSURF(SECT), CRAUNIV(COLUMN,ROW)
 2740       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Guide tube')
            LN=LN+1
 2750     CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
         SPACHEIGHT=0.0
         DO 2760 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
            SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 2760     CONTINUE
         DO 2950 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2770 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2770          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 2780 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2780          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2790 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
2790           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 2800 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
2800           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2810 V=1,(SN-1)
```

```
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                 ENDIF
              ENDIF
 2810       CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
          ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
        DO 2940 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
           IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,2820) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c     GTSECTORSURF(SECT),
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c     SPN
 2820      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,
     c     '   $ Homogenized region for spacer ',I2)
           LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,2830) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c     GTSECTORSURF(SECT),
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c     SPN
 2830      FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I4,
     c     '   $ Homogenized region for spacer ',I2)
           LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c     SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
           WRITE(30,2840) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c     (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c     GTSECTORSURF(SECT),
     c     (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
```

```
     c       SPN
 2840        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,2850) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, CRAUNIV(COLUMN,ROW),
     c       SPN
 2850        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF))) THEN
             WRITE(30,2860) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c       CRAUNIV(COLUMN,ROW), SPN
 2860        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,2870) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW), SPN
 2870        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
```

```
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2880) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
 2880        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2890) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
 2890        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2900) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
 2900        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2910) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
 2910        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
             WRITE(30,2920) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c       CRAUNIV(COLUMN,ROW)
 2920        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
```

```
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2930) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c       CRAUNIV(COLUMN,ROW)
 2930        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ENDIF
 2940        CONTINUE
 2950     CONTINUE
*  Write the moderator inside of the GT in the CR universe
         DO 3000 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
            IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c       SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this CR universe.
               WRITE(30,2960) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRCLADORSURF, (-1*CRCLADTOPSURF),
     c         CRCLADBOTTOMSURF,
     c         CRAUNIV(COLUMN,ROW)
 2960          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
               WRITE(30,2970) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         CRCLADORSURF, (-1*CRCLADTOPSURF),
     c         GTSECTBOTSURF(GTSECT),
     c         CRAUNIV(COLUMN,ROW)
 2970          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c       SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c       SURFVALUESPEC(CRCLADBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c       SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
               WRITE(30,2980) LN, BMODML, (-1*MODDENSITY),
```

```
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          CRCLADBOTTOMSURF,
     c          CRAUNIV(COLUMN,ROW)
 2980           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
                WRITE(30,2990) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          CRCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          CRAUNIV(COLUMN,ROW)
 2990           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ENDIF
 3000     CONTINUE
*   Determine the axial GT section which contains the lowest CR axial section
       DO 3010 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c    SURFVALUESPEC(CRCLADBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c    SURFVALUESPEC(CRCLADBOTTOMSURF))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 3010  CONTINUE
       DO 3040 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,3020) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*CRCLADBOTTOMSURF),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 3020        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,3030) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       CRAUNIV(COLUMN,ROW)
 3030        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
```

```
         ENDIF
 3040 CONTINUE
*     Write the lower end-fitting cell specification for this CR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,3050) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         CRAUNIV(COLUMN,ROW)
 3050          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,3060) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*CRLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         CRAUNIV(COLUMN,ROW)
 3060          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
               WRITE(30,3070) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         CRAUNIV(COLUMN,ROW)
 3070          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this CR universe.
            GTTOPSURF=GTSECTTOPSURF(1)
            IF ((SURFVALUESPEC(GTTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,3080) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3080          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,3090) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3090          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
               WRITE(30,3100) LN, FRUEFML(COLUMN,ROW),
```

```
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 3100           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).GE.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3110) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 3110           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).GE.
     c         SURFVALUESPEC(UEFTOPSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3120) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3120           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3130) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 3130           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF)).AND.
     c         (SURFVALUESPEC(GTTOPSURF).GT.
     c         SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c         SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c         (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c         SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3140           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3150) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
 3150           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3160) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), GTSECTIRSURF(1),
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
3160            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).EQ.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3170) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3170            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3180) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
3180            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(CRCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3190) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
3190            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3200) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          CRAUNIV(COLUMN,ROW)
3200            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                LN=LN+1
                WRITE(30,3210) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*CRCLADTOPSURF), CRCLADORSURF,
     c          (-1*GTSECTORSURF(1)), CRAUNIV(COLUMN,ROW)
3210            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
```

```
     c         ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
               LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(CRCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,3220) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), CRCLADORSURF, CRAUNIV(COLUMN,ROW)
 3220            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3230) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c           (-1*UEFTOPSURF), (-1*CRCLADORSURF),
     c           CRAUNIV(COLUMN,ROW)
 3230            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GE.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                 WRITE(30,3240) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3240            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3250) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c           (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c           CRAUNIV(COLUMN,ROW)
 3250            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
              ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                 WRITE(30,3260) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c           (-1*UEFTOPSURF), GTSECTORSURF(1), CRAUNIV(COLUMN,ROW)
 3260            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
                 LN=LN+1
                 WRITE(30,3270) LN, FRUEFML(COLUMN,ROW),
     c           (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), CRCLADTOPSURF,
     c           (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c           CRAUNIV(COLUMN,ROW)
```

```
3270          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
              WRITE(30,3275) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        GTSECTIRSURF(1), CRAUNIV(COLUMN,ROW)
3275          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
            ELSEIF ((SURFVALUESPEC(CRCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,3280) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), CRAUNIV(COLUMN,ROW)
3280          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Upper end-fitting')
              LN=LN+1
            ENDIF
      ENDIF
9300              FORMAT(T1,'M',I4,T9,' 8016.50c     -0.120',
     c                '          $ Zirc-4 Cladding')
9301                   FORMAT(T9,'24050.60c     -0.004')
7000                   FORMAT(T9,'24052.60c     -0.084')
7001                   FORMAT(T9,'24053.60c     -0.010')
7002                   FORMAT(T9,'24054.60c     -0.002')
9302                   FORMAT(T9,'26054.60c     -0.011')
7003                   FORMAT(T9,'26056.60c     -0.184')
7004                   FORMAT(T9,'26057.60c     -0.004')
7005                   FORMAT(T9,'26058.60c     -0.001')
9303                   FORMAT(T9,'40000.60c     -98.180')
9304                   FORMAT(T9,'50000.35c     -1.400')
9305              FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
     c                '          $ SS304 Cladding')
9306                   FORMAT(T9,'7014.50c      -0.100')
9307                   FORMAT(T9,'14000.50c     -0.750')
9308                   FORMAT(T9,'15031.50c     -0.045')
9309                   FORMAT(T9,'16032.50c     -0.030')
9310                   FORMAT(T9,'24050.60c     -0.793')
7006                   FORMAT(T9,'24052.60c     -15.903')
7007                   FORMAT(T9,'24053.60c     -1.838')
7008                   FORMAT(T9,'24054.60c     -0.466')
9311                   FORMAT(T9,'25055.50c     -2.000')
9312                   FORMAT(T9,'26054.60c     -3.918')
7009                   FORMAT(T9,'26056.60c     -63.156')
7010                   FORMAT(T9,'26057.60c     -1.472')
7011                   FORMAT(T9,'26058.60c     -0.200')
9313                   FORMAT(T9,'28058.60c     -6.234')
7012                   FORMAT(T9,'28060.60c     -2.465')
7013                   FORMAT(T9,'28061.60c     -0.109')
7014                   FORMAT(T9,'28062.60c     -0.350')
7015                   FORMAT(T9,'28064.60c     -0.092')
```

```
 9314                  FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
     c                       '           $ Inconel Cladding')
 9315                       FORMAT(T9,'14000.50c     -0.350')
 9316                       FORMAT(T9,'15031.50c     -0.015')
 9317                       FORMAT(T9,'16032.50c     -0.015')
 9318                       FORMAT(T9,'24050.60c     -0.793')
 7016                       FORMAT(T9,'24052.60c     -15.903')
 7017                       FORMAT(T9,'24053.60c     -1.838')
 7018                       FORMAT(T9,'24054.60c     -0.466')
 9319                       FORMAT(T9,'25055.50c     -0.350')
 9320                       FORMAT(T9,'26054.60c     -0.958')
 7019                       FORMAT(T9,'26056.60c     -15.442')
 7020                       FORMAT(T9,'26057.60c     -0.360')
 7021                       FORMAT(T9,'26058.60c     -0.049')
 9321                       FORMAT(T9,'28058.60c     -35.382')
 7022                       FORMAT(T9,'28060.60c     -13.993')
 7023                       FORMAT(T9,'28061.60c     -0.616')
 7024                       FORMAT(T9,'28062.60c     -1.989')
 7025                       FORMAT(T9,'28064.60c     -0.520')
 9322                       FORMAT(T9,'5010.50c      -1.078E-3')
 9323                       FORMAT(T9,'5011.56c      -4.925E-3')
 9324                       FORMAT(T9,'13027.50c     -0.500')
 9325                       FORMAT(T9,'22000.50c     -0.900')
 9326                       FORMAT(T9,'27059.50c     -1.000')
 9327                       FORMAT(T9,'29063.60c     -0.205')
 7026                       FORMAT(T9,'29065.60c     -0.095')
 9328                       FORMAT(T9,'41093.50c     -2.563')
 9329                       FORMAT(T9,'42000.50c     -3.050')
 9330                       FORMAT(T9,'73181.50c     -2.563')
      ENDIF
 3720    CONTINUE
 3730 CONTINUE

      RETURN
      END

************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
************************************************************************
      SUBROUTINE WESTBPR(BANKNUM, BMODML, BPCLADML, BPNODEML,
     c BPRAUNIV, BPRCLADMAT, BPRLPML, BPRUPML, COLUMN, DESNUM,
     c FRLEFML, FRUEFML, FRUREGIONML, GTAXMAT, GTAXML, GTDATA,
     c GTMAT, GTML, GTSPLIT, HOMOSPACMLNUM, LN, MN,
     c NUMOFBPRANODES, NUMOFGTAXS, NUMOFSPACERS,
     c NUMREGABOVEBPRA, ROW, SN, SYSTEMTOP, WBPRA,
```

```
      c WBPRATYPE, AL2O3B4CDENSITY, AL2O3DENSITY,
      c BOTBPNODEHEIGHT, BPDENTOGO, BPNONABSMAT,
      c BPRAXDIM, BPRDIM, BPRLPLENMAT, BPRLPLENWTS,
      c BPRPLEN, BPRUPLENMAT, BPRUPLENWTS, ENDFITHEIGHT,
      c GTAXDATA, HOMOSPACERDEN, LEFMAT, MCNPBPRAHEIGHT,
      c MODDENSITY, NONBPMATDATA, REGABOVEBPRA,
      c SPACERDIST, SPACERHEIGHT, SURFVALUESPEC, UEFMAT,
      c BANKDES, BPRABSNODE, BPRLPLENZAIDS,
      c BPRUPLENZAIDS, CURRENTSURFLABEL, SURFTYPESPEC,
      c NODEBOTTOMSURF)
*
       INTEGER BANKNUM(50,50), BGT, BMODML, BPCLADBOTTOMSURF,
      c BPCLADIRSURF, BPCLADML(50,50), BPCLADORSURF, BPCLADTOPSURF,
      c BPICIRSURF, BPICORSURF, BPIRSURF, BPLEFTOPSURF,
      c BPNODEBOTTOMSURF, BPNODEML, BPNODETOPSURF, BPNONABSMAT(20),
      c BPOCIRSURF,
      c BPOCORSURF, BPORSURF, BPRADIUS, BPRAUNIV(50,50),
      c BPRCLADMAT(20), BPRLPML(50,50), BPRUPML(50,50), C, CO,
      c COLUMN, CURRENTSURFLABEL,
      c DESNUM(50,50), FRLEFML(50,50), FRUEFML(50,50),
      c FRUREGIONML(50,50,20), GTAXMAT(20,5), GTAXML(50,50,5),
      c GTBOTSURF, GTIRSURF, GTMAT(20), GTML(50,50), GTORSURF,
      c GTSECT, GTSECTBOTSURF(5), GTSECTIRSURF(5), GTSECTORSURF(5),
      c GTSECTTOPSURF(5), GTSPLIT, GTTOPSURF, HOMOSPACMLNUM(20,15),
      c LN, MCNPNODE, MN, NODEBOTTOMSURF, NUMOFBPRANODES(20),
      c NUMOFGTAXS(20), NUMOFSPACERS(20), NUMREGABOVEBPRA, REGION,
      c REGIONBOTTOMSURF, REGIONTOPSURF, RO, ROW, SECT, SN,
      c SPACERBOTTOMSURF, SPACERTOPSURF, SPN, SYSTEMTOP,
      c TOPBPNODETOPSURF, UEFBOTTOMSURF, UEFTOPSURF, V,
      c WATERREGIONBOTTOMSURF, WATERREGIONTOPSURF, WBPRA(20),
      c WBPRATYPE(20), Z
*
       REAL AL2O3B4CDENSITY(20), AL2O3DENSITY(20),
      c BOTBPNODEHEIGHT(20), BPDENTOGO(50,50,50),
      c BPRAXDIM(20,6), BPRDIM(20,3), BPRLPLENMAT(20,2),
      c BPRLPLENWTS(20,35), BPRPLEN(20,2), BPRUPLENMAT(20,2),
      c BPRUPLENWTS(20,35), CLADRHO, CURRENTSURF,
      c ENDFITHEIGHT(20,2), GTAXDATA(20,4,5), GTDATA(20,4),
      c HOMOSPACERDEN(20,15),
      c LEFMAT(20,2), MCNPBPRAHEIGHT(20,50), MODDENSITY,
      c NONBPMATDATA(20,2), REGABOVEBPRA(20,3), SPACERDIST(20,15),
      c SPACERHEIGHT(20,15), SPACHEIGHT, SURFVALUESPEC(500),
      c TOTBPHEIGHT, UEFMAT(20,2)
*
       CHARACTER BANKDES(20)*5, BPRABSNODE(20,50)*1,
      c BPRLPLENZAIDS(20,35)*9, BPRUPLENZAIDS(20,35)*9,
      c SURFTYPESPEC(500)*2
*
       LOGICAL BPRLPMLUNIQUE, BPRUPMLUNIQUE, CLADMLUNIQUE, LEAVE
*
       IF (GTSPLIT.EQ.1) THEN
          IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.1) THEN
          DO 110 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*      Define the upper end-fitting bottom surface.
```

```
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 10 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
   10           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   UEFBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*     Define the upper end-fitting top surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 20 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
   20           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   UEFTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP node bounding surfaces.
             IF (MCNPNODE.EQ.1) THEN
                TOTBPHEIGHT=0.0
                DO 30 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                   TOTBPHEIGHT=TOTBPHEIGHT+
     c             MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
   30           CONTINUE
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c          TOTBPHEIGHT
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 40 V=1,(SN-1)
```

```
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 40            CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  TOPBPNODETOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  TOPBPNODETOPSURF=CURRENTSURFLABEL
               ENDIF
               BPNODETOPSURF=TOPBPNODETOPSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 50 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 50            CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 60 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 60            CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPRADIUS=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
```

```
                  BPRADIUS=CURRENTSURFLABEL
               ENDIF
            ELSEIF (MCNPNODE.NE.1) THEN
               BPNODETOPSURF=BPNODEBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 70 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
   70          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ENDIF
         IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*     Write the BP node cells in this BPR universe.
         IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
            WRITE(30,80) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPRADIUS),
     c      (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
   80       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,90) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPRADIUS),
     c      (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
   90       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,100) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPRADIUS),
     c      (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
```

```
     c       BPRAUNIV(COLUMN,ROW), MCNPNODE
  100        FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
             LN=LN+1
             BPNODEML=BPNODEML+1
           ENDIF
           ENDIF
  110      CONTINUE
*     Define the BPR cladding inner radius.
                CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 120 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
  120           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BPR cladding outer radius.
                CURRENTSURF=BPRDIM(BANKNUM(COLUMN,ROW),3)
                CURRENTSURFLABEL=0
                DO 130 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
  130           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BPR cladding top surface.
                CURRENTSURF=TOTBPHEIGHT+
     c          BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c          BPRPLEN(BANKNUM(COLUMN,ROW),1)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
```

```
                DO 140 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
  140           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BPR cladding bottom surface.
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c          BPRPLEN(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 150 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
  150           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*     Write the BP-to-cladding gap cell in this BPR universe.
              WRITE(30,160) LN, (-1*BPCLADIRSURF), BPRADIUS,
     c        (-1*TOPBPNODETOPSURF),
     c        BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
  160         FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '   $ Burnable poison-to-cladding gap')
              LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
              CLADMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 180 RO=1,(ROW-1)
                    DO 170 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
```

```
                  IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                     IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c               BPRCLADMAT(BANKNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
                  ENDIF
 170           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
 180        CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
               DO 200 RO=ROW,ROW
                  DO 190 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 190                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 200              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 220 RO=1,(ROW-1)
                  DO 210 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 210                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 220              CONTINUE
```

```
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 240 RO=1,1
                  DO 230 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
  230             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  240          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 250 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
  250                CONTINUE
                  ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c            .EQ.2) THEN
                     DO 260 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
                           WRITE(200,7006)
                           WRITE(200,7007)
                           WRITE(200,7008)
```

```
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
260                CONTINUE
                 ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c           .EQ.3) THEN
                   DO 270 C=1,2
                     IF (C.EQ.1) THEN
                       WRITE(200,9314) BPCLADML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                       WRITE(200,9315)
                       WRITE(200,9316)
                       WRITE(200,9317)
                       WRITE(200,9318)
                       WRITE(200,7016)
                       WRITE(200,7017)
                       WRITE(200,7018)
                       WRITE(200,9319)
                       WRITE(200,9320)
                       WRITE(200,7019)
                       WRITE(200,7020)
                       WRITE(200,7021)
                       WRITE(200,9321)
                       WRITE(200,7022)
                       WRITE(200,7023)
                       WRITE(200,7024)
                       WRITE(200,7025)
                       WRITE(200,9322)
                       WRITE(200,9323)
                       WRITE(200,9324)
                       WRITE(200,9325)
                       WRITE(200,9326)
                       WRITE(200,9327)
                       WRITE(200,7026)
                       WRITE(200,9328)
                       WRITE(200,9329)
                       WRITE(200,9330)
                     ENDIF
270                CONTINUE
                 ENDIF
               MN=MN+1
             ENDIF
             IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
               CLADRHO=6.56
             ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
               CLADRHO=7.90
```

```
            ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
               CLADRHO=8.19
            ENDIF
            WRITE(30,280) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c      BPCLADIRSURF,
     c      (-1*BPCLADORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
  280       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR cladding')
            LN=LN+1
*     Write the BPR upper plenum cell in this BPR universe.
*     Determine if the BPR upper plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the upper plenum material specification label.
            BPRUPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 300 RO=1,(ROW-1)
                  DO 290 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
  290             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  300          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 320 RO=ROW,ROW
                     DO 310 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
  310                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  320             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 340 RO=1,(ROW-1)
                  DO 330 CO=1,50
```

```
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        BPRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
  330          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
  340       CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 360 RO=1,1
               DO 350 CO=1,(COLUMN-1)
                  IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        BPRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
  350          CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
  360       CONTINUE
         ENDIF
      IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c SURFVALUESPEC(UEFTOPSURF)) THEN
         IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
            BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
            DO 390 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
               IF (C.EQ.1) THEN
                  WRITE(200,370) BPRUPML(COLUMN,ROW),
     c            BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c            (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
  370             FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c            '   $ Burnable Poison Rod Upper Plenum')
               ELSE
                  WRITE(200,380)
     c            BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c            (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
  380             FORMAT(T9,A9,3X,G14.6)
               ENDIF
  390          CONTINUE
            MN=MN+1
         ENDIF
         WRITE(30,400) LN, BPRUPML(COLUMN,ROW),
```

```
     c       (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c       TOPBPNODETOPSURF,
     c       (-1*BPCLADTOPSURF), (-1*BPCLADIRSURF),
     c       BPRAUNIV(COLUMN,ROW)
  400        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
             LN=LN+1
          ENDIF
*      Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*      Determine if the BPR lower plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the lower plenum material specification label.
          BPRLPMLUNIQUE=.TRUE.
          LEAVE=.FALSE.
          IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
             DO 420 RO=1,(ROW-1)
                DO 410 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                         BPRLPMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                         EXIT
                      ENDIF
                   ENDIF
  410           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
  420        CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 440 RO=ROW,ROW
                   DO 430 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
                            BPRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
  430              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
  440           CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 460 RO=1,(ROW-1)
                DO 450 CO=1,50
                   IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                   BANKNUM(CO,RO)) THEN
```

```
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
  450                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
  460             CONTINUE
                ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                   DO 480 RO=1,1
                      DO 470 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                               BPRLPMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
  470                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
  480              CONTINUE
                ENDIF
                IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
                   BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                   DO 510 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                      IF (C.EQ.1) THEN
                         WRITE(200,490) BPRLPML(COLUMN,ROW),
     c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
  490                    FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                   '   $ Burnable Poison Rod Lower Plenum')
                      ELSE
                         WRITE(200,500)
     c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
  500                    FORMAT(T9,A9,3X,G14.6)
                      ENDIF
  510                CONTINUE
                   MN=MN+1
                ENDIF
                WRITE(30,520) LN, BPRLPML(COLUMN,ROW),
     c          (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c          (-1*BPNODEBOTTOMSURF), (-1*BPCLADIRSURF),
     c          BPRAUNIV(COLUMN,ROW)
  520           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
```

```
            LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
               DO 560 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 530 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 530              CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,540) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 540              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,550) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 550              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 560           CONTINUE
     DO 610 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
```

```
*     Define the GT section top surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
                IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 570 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 570            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTTOPSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section bottom surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
                CURRENTSURFLABEL=0
                DO 580 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 580            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTBOTSURF(SECT)=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
                ENDIF
*     Define the GT section outer radius surface.
                CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
                CURRENTSURFLABEL=0
                DO 590 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 590            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTORSURF(SECT)=SN
```

```
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 600 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 600           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTIRSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTIRSURF(SECT)=CURRENTSURFLABEL
               ENDIF
 610  CONTINUE
*     Write the GT material cell
         DO 740 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 630 RO=1,(ROW-1)
                  DO 620 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 620              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
```

```
                  EXIT
               ENDIF
630          CONTINUE
             IF (LEAVE.EQ..FALSE.) THEN
                DO 650 RO=ROW,ROW
                   DO 640 CO=1,(COLUMN-1)
                      IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
                         IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTMAT(DESNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                            EXIT
                         ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                   GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                            EXIT
                         ENDIF
                      ENDIF
640                CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
650             CONTINUE
             ENDIF
          ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
             DO 670 RO=1,(ROW-1)
                DO 660 CO=1,50
                   IF ((DESNUM(CO,RO).NE.0).AND.
     c                (BANKNUM(CO,RO).EQ.0)) THEN
                      IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTMAT(DESNUM(CO,RO))) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                         EXIT
                      ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                         CLADMLUNIQUE=.FALSE.
                         LEAVE=.TRUE.
                         GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                         EXIT
                      ENDIF
                   ENDIF
660             CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                   EXIT
                ENDIF
670          CONTINUE
          ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
             DO 690 RO=1,1
```

```
                  DO 680 CO=1,(COLUMN-1)
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
  680             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
  690          CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                  IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 700 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
  700                CONTINUE
                  ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c            .EQ.2) THEN
                     DO 710 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
                           WRITE(200,9308)
                           WRITE(200,9309)
                           WRITE(200,9310)
```

```
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
710                CONTINUE
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
   c             .EQ.3) THEN
                   DO 720 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9315)
                         WRITE(200,9316)
                         WRITE(200,9317)
                         WRITE(200,9318)
                         WRITE(200,7016)
                         WRITE(200,7017)
                         WRITE(200,7018)
                         WRITE(200,9319)
                         WRITE(200,9320)
                         WRITE(200,7019)
                         WRITE(200,7020)
                         WRITE(200,7021)
                         WRITE(200,9321)
                         WRITE(200,7022)
                         WRITE(200,7023)
                         WRITE(200,7024)
                         WRITE(200,7025)
                         WRITE(200,9322)
                         WRITE(200,9323)
                         WRITE(200,9324)
                         WRITE(200,9325)
                         WRITE(200,9326)
                         WRITE(200,9327)
                         WRITE(200,7026)
                         WRITE(200,9328)
                         WRITE(200,9329)
                         WRITE(200,9330)
                      ENDIF
720                CONTINUE
                 ENDIF
              MN=MN+1
           ENDIF
           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
```

```
                CLADRHO=6.56
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                CLADRHO=7.90
              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                CLADRHO=8.19
              ENDIF
              WRITE(30,730) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c        GTSECTIRSURF(SECT),
     c        (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c        GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
  730         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,'   $ Guide tube')
              LN=LN+1
  740       CONTINUE
*       Loop through the spacer and moderator regions along the axial
*       length of the GT (from top to bottom).
            SPACHEIGHT=0.0
            DO 750 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
              SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
  750       CONTINUE
            DO 940 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*       Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                SPACERTOPSURF=UEFBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 760 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
  760           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 770 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
  770           CONTINUE
```

```
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c   NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 780 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 780        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 790 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 790        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
```

```
                DO 800 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
  800           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
          DO 930 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
             IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,810) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
  810       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,820) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
  820       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,830) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
```

```
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
  830        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,840) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
  840        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF))) THEN
             WRITE(30,850) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c       BPRAUNIV(COLUMN,ROW), SPN
  850        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,860) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), SPN
  860        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
```

```
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,870) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
  870        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,880) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
  880        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,890) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
  890        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,900) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
  900        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
             WRITE(30,910) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c       BPRAUNIV(COLUMN,ROW)
  910        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
```

```
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,920) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
  920       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ENDIF
  930       CONTINUE
  940    CONTINUE
*  Write the moderator inside of the GT in the BPR universe
         DO 990 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
            IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c      SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
               WRITE(30,950) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         BPCLADORSURF, (-1*BPCLADTOPSURF),
     c         BPCLADBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW)
  950          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c      SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c      SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
               WRITE(30,960) LN, BMODML, (-1*MODDENSITY),
     c         (-1*GTSECTIRSURF(GTSECT)),
     c         BPCLADORSURF, (-1*BPCLADTOPSURF),
     c         GTSECTBOTSURF(GTSECT),
     c         BPRAUNIV(COLUMN,ROW)
  960          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Borated moderator inside guide tube')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c      SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
```

```
                WRITE(30,970) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          BPCLADBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
  970           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,980) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPCLADORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
  980           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ENDIF
  990       CONTINUE
*   Determine the axial GT section which contains the lowest BPR axial
section
        DO 1000 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c    (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c    SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 1000   CONTINUE
        DO 1030 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,1010) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*BPCLADBOTTOMSURF),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 1010        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,1020) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 1020        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
```

```
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 1030 CONTINUE
*     Define the lower end-fitting top surface.
              CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 1040 V=1,(SN-1)
                 IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                       CURRENTSURFLABEL=V
                       EXIT
                    ENDIF
                 ENDIF
 1040            CONTINUE
              IF (CURRENTSURFLABEL.EQ.0) THEN
                 BPLEFTOPSURF=SN
                 SURFTYPESPEC(SN)='PZ'
                 SURFVALUESPEC(SN)=CURRENTSURF
                 SN=SN+1
              ELSE
                 BPLEFTOPSURF=CURRENTSURFLABEL
              ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
             GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
             IF (SURFVALUESPEC(GTBOTSURF).GE.
     c       ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
                WRITE(30,1050) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c          BPRAUNIV(COLUMN,ROW)
 1050           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c          '   $ Lower end-fitting')
                LN=LN+1
             ELSE
                WRITE(30,1060) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c          GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c          BPRAUNIV(COLUMN,ROW)
 1060           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
                LN=LN+1
                WRITE(30,1070) LN, FRLEFML(COLUMN,ROW),
     c          (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c          (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c          (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c          BPRAUNIV(COLUMN,ROW)
 1070           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
                LN=LN+1
             ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
             IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
```

```
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1080) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 1080           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1090) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1090           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1100) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 1100           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c        SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,1110) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
 1110           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1120) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1120           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1130) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1130           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c        SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
```

```
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c       SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,1140) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 1140           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
                WRITE(30,1150) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 1150           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,1160) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 1160           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1170) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1170           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '    $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1180) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 1180           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'    $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
```

```
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,1190) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 1190           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1200) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1200           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,1210) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 1210           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,1220) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 1220           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
             ENDIF
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
*******************************************************************
```

```
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
         ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.2) THEN
         DO 1390 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1230 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1230          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 1240 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 1240          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
```

```
                DO 1250 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1250           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPICIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPICIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the inner BPR cladding outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 1260 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1260           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPICORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPICORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP absorber inner radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
                CURRENTSURFLABEL=0
                DO 1270 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 1270           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPIRSURF=CURRENTSURFLABEL
                ENDIF
```

```
*     Define the BP absorber outer radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
            CURRENTSURFLABEL=0
            DO 1280 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
1280        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPORSURF=CURRENTSURFLABEL
            ENDIF
*     Define the outer BPR cladding inner radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
            CURRENTSURFLABEL=0
            DO 1290 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
1290        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPOCIRSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               BPOCIRSURF=CURRENTSURFLABEL
            ENDIF
*     Define the outer BPR cladding outer radius.
            CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
            CURRENTSURFLABEL=0
            DO 1300 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'CZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
1300        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPOCORSURF=SN
               SURFTYPESPEC(SN)='CZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
```

```
               ELSE
                  BPOCORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP node bounding surfaces.
            IF (MCNPNODE.EQ.1) THEN
               TOTBPHEIGHT=0.0
               DO 1310 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                  TOTBPHEIGHT=TOTBPHEIGHT+
     c            MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 1310          CONTINUE
               CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c         TOTBPHEIGHT
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1320 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1320          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  TOPBPNODETOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  TOPBPNODETOPSURF=CURRENTSURFLABEL
               ENDIF
               BPNODETOPSURF=TOPBPNODETOPSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1330 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1330          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
```

```
               ENDIF
             ELSEIF (MCNPNODE.NE.1) THEN
               BPNODETOPSURF=BPNODEBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1350 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1350          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
             ENDIF
           IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)) THEN
*       Write the BP node cells in this BPR universe.
           IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c        (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
              WRITE(30,1360) LN, BPNODEML,
     c        (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
 1360         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
              LN=LN+1
              BPNODEML=BPNODEML+1
           ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c        (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
              WRITE(30,1370) LN, BPNODEML,
     c        (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
 1370         FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c        1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
              LN=LN+1
              BPNODEML=BPNODEML+1
           ELSE
              WRITE(30,1380) LN, BPNODEML,
     c        (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c        BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW), MCNPNODE
```

```
 1380        FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
             LN=LN+1
             BPNODEML=BPNODEML+1
           ENDIF
           ENDIF
 1390      CONTINUE
*     Define the BPR cladding top surface.
             CURRENTSURF=TOTBPHEIGHT+
     c       BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c       BPRPLEN(BANKNUM(COLUMN,ROW),1)
             IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
             ENDIF
             CURRENTSURFLABEL=0
             DO 1400 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 1400        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPCLADTOPSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPCLADTOPSURF=CURRENTSURFLABEL
             ENDIF
*     Define the BPR cladding bottom surface.
             CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c       BPRPLEN(BANKNUM(COLUMN,ROW),2)
             CURRENTSURFLABEL=0
             DO 1410 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
           IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 1410        CONTINUE
             IF (CURRENTSURFLABEL.EQ.0) THEN
                BPCLADBOTTOMSURF=SN
                SURFTYPESPEC(SN)='PZ'
                SURFVALUESPEC(SN)=CURRENTSURF
                SN=SN+1
             ELSE
                BPCLADBOTTOMSURF=CURRENTSURFLABEL
             ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
           WRITE(30,1420) LN, (-1*BPIRSURF), BPICORSURF,
     c     (-1*TOPBPNODETOPSURF),
```

```
     c       BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1420        FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Burnable poison-to-cladding gap')
             LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
             WRITE(30,1430) LN, (-1*BPOCIRSURF), BPORSURF,
     c       (-1*TOPBPNODETOPSURF),
     c       BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1430        FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Burnable poison-to-cladding gap')
             LN=LN+1
*     Write the annular gap cell in this BPR universe.
             WRITE(30,1440) LN, (-1*BPICIRSURF),
     c       (-1*TOPBPNODETOPSURF),
     c       BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 1440        FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Burnable poison-to-cladding gap')
             LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
             CLADMLUNIQUE=.TRUE.
             LEAVE=.FALSE.
             IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 1460 RO=1,(ROW-1)
                   DO 1450 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   BPRCLADMAT(BANKNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 1450                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1460              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 1480 RO=ROW,ROW
                         DO 1470 CO=1,(COLUMN-1)
                            IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                               IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                         BPRCLADMAT(BANKNUM(CO,RO))) THEN
                                  CLADMLUNIQUE=.FALSE.
```

```
                                LEAVE=.TRUE.
                                BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
                          ENDIF
 1470                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 1480                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 1500 RO=1,(ROW-1)
                     DO 1490 CO=1,50
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 1490                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1500             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 1520 RO=1,1
                     DO 1510 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 1510                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1520             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
```

```
                IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                   DO 1530 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9300) BPCLADML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9301)
                         WRITE(200,7000)
                         WRITE(200,7001)
                         WRITE(200,7002)
                         WRITE(200,9302)
                         WRITE(200,7003)
                         WRITE(200,7004)
                         WRITE(200,7005)
                         WRITE(200,9303)
                         WRITE(200,9304)
                      ENDIF
1530                CONTINUE
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c          .EQ.2) THEN
                   DO 1540 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9305) BPCLADML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9306)
                         WRITE(200,9307)
                         WRITE(200,9308)
                         WRITE(200,9309)
                         WRITE(200,9310)
                         WRITE(200,7006)
                         WRITE(200,7007)
                         WRITE(200,7008)
                         WRITE(200,9311)
                         WRITE(200,9312)
                         WRITE(200,7009)
                         WRITE(200,7010)
                         WRITE(200,7011)
                         WRITE(200,9313)
                         WRITE(200,7012)
                         WRITE(200,7013)
                         WRITE(200,7014)
                         WRITE(200,7015)
                      ENDIF
1540                CONTINUE
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c          .EQ.3) THEN
                   DO 1550 C=1,2
                      IF (C.EQ.1) THEN
                         WRITE(200,9314) BPCLADML(COLUMN,ROW)
                      ELSEIF (C.EQ.2) THEN
                         WRITE(200,9315)
                         WRITE(200,9316)
                         WRITE(200,9317)
                         WRITE(200,9318)
                         WRITE(200,7016)
```

```
                              WRITE(200,7017)
                              WRITE(200,7018)
                              WRITE(200,9319)
                              WRITE(200,9320)
                              WRITE(200,7019)
                              WRITE(200,7020)
                              WRITE(200,7021)
                              WRITE(200,9321)
                              WRITE(200,7022)
                              WRITE(200,7023)
                              WRITE(200,7024)
                              WRITE(200,7025)
                              WRITE(200,9322)
                              WRITE(200,9323)
                              WRITE(200,9324)
                              WRITE(200,9325)
                              WRITE(200,9326)
                              WRITE(200,9327)
                              WRITE(200,7026)
                              WRITE(200,9328)
                              WRITE(200,9329)
                              WRITE(200,9330)
                           ENDIF
 1550                   CONTINUE
                     ENDIF
                  MN=MN+1
               ENDIF
               IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                  CLADRHO=6.56
               ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                  CLADRHO=7.90
               ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                  CLADRHO=8.19
               ENDIF
               WRITE(30,1560) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c         BPOCIRSURF,
     c         (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW)
 1560        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR cladding')
               LN=LN+1
               WRITE(30,1570) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c         BPICIRSURF,
     c         (-1*BPICORSURF), (-1*TOPBPNODETOPSURF), BPNODEBOTTOMSURF,
     c         BPRAUNIV(COLUMN,ROW)
 1570        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR cladding')
               LN=LN+1
*      Write the BPR upper plenum cell in this BPR universe.
*      Determine if the BPR upper plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the upper plenum material specification label.
               BPRUPMLUNIQUE=.TRUE.
               LEAVE=.FALSE.
```

```
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1590 RO=1,(ROW-1)
                  DO 1580 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1580             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1590          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1610 RO=ROW,ROW
                     DO 1600 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1600                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1610             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 1630 RO=1,(ROW-1)
                  DO 1620 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1620             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1630          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 1650 RO=1,1
```

```
                  DO 1640 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 1640             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1650          CONTINUE
            ENDIF
         IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
            IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
               BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
               DO 1680 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                     WRITE(200,1660) BPRUPML(COLUMN,ROW),
     c               BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1660                FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '   $ Burnable Poison Rod Upper Plenum')
                  ELSE
                     WRITE(200,1670)
     c               BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 1670                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 1680             CONTINUE
               MN=MN+1
            ENDIF
            WRITE(30,1690) LN, BPRUPML(COLUMN,ROW),
     c      (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      TOPBPNODETOPSURF,
     c      (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 1690       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
            LN=LN+1
         ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            BPRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1710 RO=1,(ROW-1)
```

```
                  DO 1700 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
1700              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
1710           CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 1730 RO=ROW,ROW
                     DO 1720 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
1720                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
1730              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 1750 RO=1,(ROW-1)
                  DO 1740 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                     BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
1740              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
1750           CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 1770 RO=1,1
                  DO 1760 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
```

```
     c                        BANKNUM(CO,RO)) THEN
                              BPRLPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 1760                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 1770             CONTINUE
               ENDIF
               IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
                  BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                  DO 1800 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,1780) BPRLPML(COLUMN,ROW),
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1780                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '   $ Burnable Poison Rod Lower Plenum')
                     ELSE
                        WRITE(200,1790)
     c                  BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                  (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 1790                   FORMAT(T9,A9,3X,G14.6)
                     ENDIF
 1800                CONTINUE
                  MN=MN+1
               ENDIF
               WRITE(30,1810) LN, BPRLPML(COLUMN,ROW),
     c         (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c         (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 1810        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
               LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                  DO 1850 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                     IF (REGION.EQ.1) THEN
                        REGIONTOPSURF=SYSTEMTOP
                        CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                  REGABOVEBPRA(REGION,1)
                     ENDIF
                     CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c               REGABOVEBPRA(REGION,1)
                     IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                        REGIONBOTTOMSURF=UEFTOPSURF
                     ELSE
```

```
                  CURRENTSURFLABEL=0
                  DO 1820 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 1820             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,1830) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 1830             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,1840) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 1840             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 1850          CONTINUE
      DO 1900 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
               IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 1860 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1860          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTTOPSURF(SECT)=SN
```

```
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section bottom surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
               CURRENTSURFLABEL=0
               DO 1870 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1870          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTBOTSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section outer radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
               CURRENTSURFLABEL=0
               DO 1880 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 1880          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTORSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 1890 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
```

```
 1890           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTSECTIRSURF(SECT)=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTSECTIRSURF(SECT)=CURRENTSURFLABEL
                ENDIF
 1900 CONTINUE
*     Write the GT material cell
          DO 2030 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 1920 RO=1,(ROW-1)
                  DO 1910 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 1910             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 1920             CONTINUE
                  IF (LEAVE.EQ..FALSE.) THEN
                     DO 1940 RO=ROW,ROW
                        DO 1930 CO=1,(COLUMN-1)
                           IF ((DESNUM(CO,RO).NE.0).AND.
     c                     (BANKNUM(CO,RO).EQ.0)) THEN
                              IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                        GTMAT(DESNUM(CO,RO))) THEN
                                 CLADMLUNIQUE=.FALSE.
                                 LEAVE=.TRUE.
                                 GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                                 EXIT
                              ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                        GTAXMAT(DESNUM(CO,RO),SECT)) THEN
```

```
                                CLADMLUNIQUE=.FALSE.
                                LEAVE=.TRUE.
                                GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                                EXIT
                             ENDIF
                          ENDIF
1930                   CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
1940                CONTINUE
                 ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 1960 RO=1,(ROW-1)
                    DO 1950 CO=1,50
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
                       ENDIF
1950                CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
1960             CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 1980 RO=1,1
                    DO 1970 CO=1,(COLUMN-1)
                       IF ((DESNUM(CO,RO).NE.0).AND.
     c                    (BANKNUM(CO,RO).EQ.0)) THEN
                          IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTMAT(DESNUM(CO,RO))) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                             EXIT
                          ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                    GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                             CLADMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                             EXIT
                          ENDIF
```

```
                  ENDIF
 1970            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 1980          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
               IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                  DO 1990 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9301)
                        WRITE(200,7000)
                        WRITE(200,7001)
                        WRITE(200,7002)
                        WRITE(200,9302)
                        WRITE(200,7003)
                        WRITE(200,7004)
                        WRITE(200,7005)
                        WRITE(200,9303)
                        WRITE(200,9304)
                     ENDIF
 1990             CONTINUE
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c           .EQ.2) THEN
                  DO 2000 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9306)
                        WRITE(200,9307)
                        WRITE(200,9308)
                        WRITE(200,9309)
                        WRITE(200,9310)
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
 2000             CONTINUE
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
```

```
     c             .EQ.3) THEN
                      DO 2010 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9315)
                            WRITE(200,9316)
                            WRITE(200,9317)
                            WRITE(200,9318)
                            WRITE(200,7016)
                            WRITE(200,7017)
                            WRITE(200,7018)
                            WRITE(200,9319)
                            WRITE(200,9320)
                            WRITE(200,7019)
                            WRITE(200,7020)
                            WRITE(200,7021)
                            WRITE(200,9321)
                            WRITE(200,7022)
                            WRITE(200,7023)
                            WRITE(200,7024)
                            WRITE(200,7025)
                            WRITE(200,9322)
                            WRITE(200,9323)
                            WRITE(200,9324)
                            WRITE(200,9325)
                            WRITE(200,9326)
                            WRITE(200,9327)
                            WRITE(200,7026)
                            WRITE(200,9328)
                            WRITE(200,9329)
                            WRITE(200,9330)
                         ENDIF
 2010                 CONTINUE
                   ENDIF
                MN=MN+1
             ENDIF
             IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                CLADRHO=6.56
             ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                CLADRHO=7.90
             ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                CLADRHO=8.19
             ENDIF
             WRITE(30,2020) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c       GTSECTIRSURF(SECT),
     c       (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c       GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
 2020        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Guide tube')
             LN=LN+1
 2030     CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
```

```
          SPACHEIGHT=0.0
          DO 2040 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
             SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 2040     CONTINUE
          DO 2230 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2050 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2050          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 2060 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2060          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 2070 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
```

```
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2070        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
            WATERREGIONTOPSURF=SPACERBOTTOMSURF
            CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
            CURRENTSURFLABEL=0
            DO 2080 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2080        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               WATERREGIONBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
         ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
            SPACERTOPSURF=WATERREGIONBOTTOMSURF
            CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c      SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
            CURRENTSURFLABEL=0
            DO 2090 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                  CURRENTSURFLABEL=V
                  EXIT
               ENDIF
            ENDIF
2090        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               SPACERBOTTOMSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
            ELSE
               SPACERBOTTOMSURF=CURRENTSURFLABEL
            ENDIF
```

```
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
              ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
            DO 2220 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2100) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c        SPN
 2100         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2110) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c        SPN
 2110         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2120) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c        SPN
 2120         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I4,
     c        '   $ Homogenized region for spacer ',I2)
              LN=LN+1
               ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c        SURFVALUESPEC(SPACERTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c        SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
              WRITE(30,2130) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c        (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c        GTSECTORSURF(SECT),
     c        (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c        SPN
 2130         FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
```

```
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF))) THEN
             WRITE(30,2140) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c       BPRAUNIV(COLUMN,ROW), SPN
 2140        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,2150) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), SPN
 2150        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2160) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 2160        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2170) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
```

```
     c       BPRAUNIV(COLUMN,ROW)
 2170        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2180) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 2180        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2190) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 2190        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
             WRITE(30,2200) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c       BPRAUNIV(COLUMN,ROW)
 2200        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,2210) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 2210        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ENDIF
 2220        CONTINUE
```

```
 2230    CONTINUE
*  Write the moderator inside of the GT in the BPR universe
        DO 2280 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
           IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c     SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c     SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
              WRITE(30,2240) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTSECTIRSURF(GTSECT)),
     c        BPOCORSURF, (-1*BPCLADTOPSURF),
     c        BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2240        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '   $ Borated moderator inside guide tube')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c     SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c     SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c     SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
              WRITE(30,2250) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTSECTIRSURF(GTSECT)),
     c        BPOCORSURF, (-1*BPCLADTOPSURF),
     c        GTSECTBOTSURF(GTSECT),
     c        BPRAUNIV(COLUMN,ROW)
 2250        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '   $ Borated moderator inside guide tube')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c     SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c     SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c     SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
              WRITE(30,2260) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTSECTIRSURF(GTSECT)),
     c        BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c        BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 2260        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,
     c        '   $ Borated moderator inside guide tube')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c     SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c     (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c     SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
              WRITE(30,2270) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTSECTIRSURF(GTSECT)),
     c        BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
```

```
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 2270           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
               LN=LN+1
            ENDIF
 2280    CONTINUE
*   Determine the axial GT section which contains the lowest BPR axial
section
      DO 2290 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
         IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c   SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c   (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c   SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
            BGT=GTSECT
            EXIT
         ENDIF
 2290 CONTINUE
      DO 2320 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
         IF (GTSECT.EQ.BGT) THEN
            WRITE(30,2300) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      (-1*BPCLADBOTTOMSURF),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 2300       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Borated moderator inside guide tube')
            LN=LN+1
         ELSE
            WRITE(30,2310) LN, BMODML, (-1*MODDENSITY),
     c      (-1*GTSECTIRSURF(GTSECT)),
     c      (-1*GTSECTTOPSURF(GTSECT)),
     c      GTSECTBOTSURF(GTSECT),
     c      BPRAUNIV(COLUMN,ROW)
 2310       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,
     c      '   $ Borated moderator inside guide tube')
            LN=LN+1
         ENDIF
 2320 CONTINUE
*     Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2330 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 2330          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  BPLEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,2340) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
 2340          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,2350) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         BPRAUNIV(COLUMN,ROW)
 2350          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
               WRITE(30,2360) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         BPRAUNIV(COLUMN,ROW)
 2360          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
            IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,2370) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 2370          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,2380) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
```

```
2380          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,2390) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c        BPRAUNIV(COLUMN,ROW)
2390          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,2400) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
2400          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,2410) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c        (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c        BPRAUNIV(COLUMN,ROW)
2410          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,2420) LN, BMODML,
     c        (-1*MODDENSITY), BPCLADTOPSURF,
     c        (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c        BPRAUNIV(COLUMN,ROW)
2420          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,2430) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTSECTORSURF(1),
     c        BPRAUNIV(COLUMN,ROW)
2430          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
            IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
              WRITE(30,2440) LN, BMODML,
     c        (-1*MODDENSITY), BPCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c        BPRAUNIV(COLUMN,ROW)
2440          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
```

```
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
             ENDIF
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
               WRITE(30,2450) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1),
     c         BPRAUNIV(COLUMN,ROW)
 2450        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,2460) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c         GTSECTTOPSURF(1),
     c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c         BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 2460        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         1X,I4,' IMP:N=1 U=',I3,
     c         '   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,2470) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 2470        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
             ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c       SURFVALUESPEC(UEFTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
               WRITE(30,2480) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTSECTORSURF(1),
     c         BPRAUNIV(COLUMN,ROW)
 2480        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,2490) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c         GTSECTTOPSURF(1),
     c         (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c         BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 2490        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         1X,I4,' IMP:N=1 U=',I3,
```

```
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,2500) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 2500           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,2510) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 2510           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
*************************************************************************
          ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3) THEN
          DO 2680 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
              CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c        ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
              CURRENTSURFLABEL=0
              DO 2520 V=1,(SN-1)
                IF (SURFTYPESPEC(V).EQ.'PZ') THEN
```

```
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 2520            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    UEFBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    UEFBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
*      Define the upper end-fitting top surface.
                 CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c           ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                 CURRENTSURFLABEL=0
                 DO 2530 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 2530            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    UEFTOPSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    UEFTOPSURF=CURRENTSURFLABEL
                 ENDIF
*      Define the inner BPR cladding inner radius.
                 CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
                 CURRENTSURFLABEL=0
                 DO 2540 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'CZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                          CURRENTSURFLABEL=V
                          EXIT
                       ENDIF
                    ENDIF
 2540            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    BPICIRSURF=SN
                    SURFTYPESPEC(SN)='CZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    BPICIRSURF=CURRENTSURFLABEL
                 ENDIF
```

```
*     Define the inner BPR cladding outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 2550 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2550           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPICORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPICORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP absorber inner radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
                CURRENTSURFLABEL=0
                DO 2560 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2560           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP absorber outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
                CURRENTSURFLABEL=0
                DO 2570 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2570           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
```

```
                ELSE
                   BPORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the outer BPR cladding inner radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
                CURRENTSURFLABEL=0
                DO 2580 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2580           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPOCIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPOCIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the outer BPR cladding outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
                CURRENTSURFLABEL=0
                DO 2590 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 2590           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPOCORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPOCORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP node bounding surfaces.
              IF (MCNPNODE.EQ.1) THEN
                TOTBPHEIGHT=0.0
                DO 2600 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                   TOTBPHEIGHT=TOTBPHEIGHT+
     c             MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 2600           CONTINUE
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c          TOTBPHEIGHT
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
```

```
                DO 2610 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
2610            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   TOPBPNODETOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   TOPBPNODETOPSURF=CURRENTSURFLABEL
                ENDIF
                BPNODETOPSURF=TOPBPNODETOPSURF
                CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c          MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2620 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
2620            CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPNODEBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPNODEBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
             ELSEIF (MCNPNODE.NE.1) THEN
                BPNODETOPSURF=BPNODEBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c          MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 2640 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
```

```
 2640          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ENDIF
         IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*     Write the BP node cells in this BPR universe.
         IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
            WRITE(30,2650) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 2650       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,2660) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 2660       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,2670) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 2670       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ENDIF
         ENDIF
 2680     CONTINUE
*     Define the BPR cladding top surface.
            CURRENTSURF=TOTBPHEIGHT+
     c      BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c      BPRPLEN(BANKNUM(COLUMN,ROW),1)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 2690 V=1,(SN-1)
```

```
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2690          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BPR cladding bottom surface.
               CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c         BPRPLEN(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 2700 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 2700          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
               WRITE(30,2710) LN, (-1*BPIRSURF), BPICORSURF,
     c         (-1*TOPBPNODETOPSURF),
     c         BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2710          FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Burnable poison-to-cladding gap')
               LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
               WRITE(30,2720) LN, (-1*BPOCIRSURF), BPORSURF,
     c         (-1*TOPBPNODETOPSURF),
     c         BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2720          FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Burnable poison-to-cladding gap')
               LN=LN+1
*     Write the annular water cell in this BPR universe.
               WRITE(30,2730) LN, BMODML,
     c         (-1*MODDENSITY), (-1*BPICIRSURF),
```

```
     c       (-1*TOPBPNODETOPSURF),
     c       BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 2730        FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Burnable poison-to-cladding gap')
             LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
             CLADMLUNIQUE=.TRUE.
             LEAVE=.FALSE.
             IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 2750 RO=1,(ROW-1)
                   DO 2740 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   BPRCLADMAT(BANKNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 2740              CONTINUE
                   IF (LEAVE.EQ..TRUE.) THEN
                      EXIT
                   ENDIF
 2750           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                   DO 2770 RO=ROW,ROW
                      DO 2760 CO=1,(COLUMN-1)
                         IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                            IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                      BPRCLADMAT(BANKNUM(CO,RO))) THEN
                               CLADMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
                         ENDIF
 2760                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 2770                 CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 2790 RO=1,(ROW-1)
                   DO 2780 CO=1,50
```

```
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2780             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2790          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2810 RO=1,1
                  DO 2800 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 2800             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 2810          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 2820 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
```

```
                  ENDIF
2820              CONTINUE
                 ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c           .EQ.2) THEN
                    DO 2830 C=1,2
                       IF (C.EQ.1) THEN
                          WRITE(200,9305) BPCLADML(COLUMN,ROW)
                       ELSEIF (C.EQ.2) THEN
                          WRITE(200,9306)
                          WRITE(200,9307)
                          WRITE(200,9308)
                          WRITE(200,9309)
                          WRITE(200,9310)
                          WRITE(200,7006)
                          WRITE(200,7007)
                          WRITE(200,7008)
                          WRITE(200,9311)
                          WRITE(200,9312)
                          WRITE(200,7009)
                          WRITE(200,7010)
                          WRITE(200,7011)
                          WRITE(200,9313)
                          WRITE(200,7012)
                          WRITE(200,7013)
                          WRITE(200,7014)
                          WRITE(200,7015)
                       ENDIF
2830                CONTINUE
                   ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c             .EQ.3) THEN
                      DO 2840 C=1,2
                         IF (C.EQ.1) THEN
                            WRITE(200,9314) BPCLADML(COLUMN,ROW)
                         ELSEIF (C.EQ.2) THEN
                            WRITE(200,9315)
                            WRITE(200,9316)
                            WRITE(200,9317)
                            WRITE(200,9318)
                            WRITE(200,7016)
                            WRITE(200,7017)
                            WRITE(200,7018)
                            WRITE(200,9319)
                            WRITE(200,9320)
                            WRITE(200,7019)
                            WRITE(200,7020)
                            WRITE(200,7021)
                            WRITE(200,9321)
                            WRITE(200,7022)
                            WRITE(200,7023)
                            WRITE(200,7024)
                            WRITE(200,7025)
                            WRITE(200,9322)
                            WRITE(200,9323)
                            WRITE(200,9324)
```

```
                              WRITE(200,9325)
                              WRITE(200,9326)
                              WRITE(200,9327)
                              WRITE(200,7026)
                              WRITE(200,9328)
                              WRITE(200,9329)
                              WRITE(200,9330)
                           ENDIF
 2840                   CONTINUE
                      ENDIF
                   MN=MN+1
                ENDIF
                IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                   CLADRHO=6.56
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                   CLADRHO=7.90
                ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                   CLADRHO=8.19
                ENDIF
                WRITE(30,2850) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c          BPOCIRSURF,
     c          (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
 2850           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ BPR cladding')
                LN=LN+1
                WRITE(30,2860) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c          BPICIRSURF,
     c          (-1*BPICORSURF), (-1*TOPBPNODETOPSURF),
     c          BPNODEBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
 2860           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ BPR cladding')
                LN=LN+1
*      Write the BPR upper plenum cell in this BPR universe.
*      Determine if the BPR upper plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the upper plenum material specification label.
                BPRUPMLUNIQUE=.TRUE.
                LEAVE=.FALSE.
                IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                   DO 2880 RO=1,(ROW-1)
                      DO 2870 CO=1,50
                         IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                               BPRUPMLUNIQUE=.FALSE.
                               LEAVE=.TRUE.
                               BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                               EXIT
                            ENDIF
                         ENDIF
 2870                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
```

```
                  EXIT
               ENDIF
2880        CONTINUE
            IF (LEAVE.EQ..FALSE.) THEN
               DO 2900 RO=ROW,ROW
                  DO 2890 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
2890                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
2900              CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 2920 RO=1,(ROW-1)
                  DO 2910 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
2910                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
2920              CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 2940 RO=1,1
                  DO 2930 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRUPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
2930                 CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
```

```
 2940          CONTINUE
              ENDIF
            IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
              IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
                BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                DO 2970 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                  IF (C.EQ.1) THEN
                    WRITE(200,2950) BPRUPML(COLUMN,ROW),
     c              BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c              (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2950               FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c              '   $ Burnable Poison Rod Upper Plenum')
                  ELSE
                    WRITE(200,2960)
     c              BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c              (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 2960               FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 2970           CONTINUE
              MN=MN+1
            ENDIF
            WRITE(30,2980) LN, BPRUPML(COLUMN,ROW),
     c      (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c      TOPBPNODETOPSURF,
     c      (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
     c      BPRAUNIV(COLUMN,ROW)
 2980       FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
            LN=LN+1
          ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
            BPRLPMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 3000 RO=1,(ROW-1)
                DO 2990 CO=1,50
                  IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                      BPRLPMLUNIQUE=.FALSE.
                      LEAVE=.TRUE.
                      BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                      EXIT
                    ENDIF
                  ENDIF
 2990           CONTINUE
                IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
                ENDIF
 3000         CONTINUE
```

```
            IF (LEAVE.EQ..FALSE.) THEN
               DO 3020 RO=ROW,ROW
                  DO 3010 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3010             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3020          CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 3040 RO=1,(ROW-1)
                  DO 3030 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3030             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3040          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 3060 RO=1,1
                  DO 3050 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                           BPRLPMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 3050             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3060          CONTINUE
            ENDIF
            IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
```

```
                BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                   DO 3090 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                      IF (C.EQ.1) THEN
                         WRITE(200,3070) BPRLPML(COLUMN,ROW),
     c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 3070                    FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                   '   $ Burnable Poison Rod Lower Plenum')
                      ELSE
                         WRITE(200,3080)
     c                   BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 3080                    FORMAT(T9,A9,3X,G14.6)
                      ENDIF
 3090                CONTINUE
                  MN=MN+1
               ENDIF
               WRITE(30,3100) LN, BPRLPML(COLUMN,ROW),
     c         (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c         (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 3100        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
               LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                  DO 3140 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                   IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                REGABOVEBPRA(REGION,1)
                   ENDIF
                   CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c             REGABOVEBPRA(REGION,1)
                   IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                      REGIONBOTTOMSURF=UEFTOPSURF
                   ELSE
                   CURRENTSURFLABEL=0
                   DO 3110 V=1,(SN-1)
                      IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                 IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                            CURRENTSURFLABEL=V
                            EXIT
                         ENDIF
                      ENDIF
 3110               CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                      REGIONBOTTOMSURF=SN
                      SURFTYPESPEC(SN)='PZ'
                      SURFVALUESPEC(SN)=CURRENTSURF
                      SN=SN+1
```

```
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,3120) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 3120             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,3130) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 3130             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 3140          CONTINUE
      DO 3190 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Define the GT section top surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),3,SECT)
               IF (CURRENTSURF.GT.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 3150 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3150          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTTOPSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTTOPSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section bottom surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),4,SECT)
               CURRENTSURFLABEL=0
               DO 3160 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
```

```
                  ENDIF
               ENDIF
 3160          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTBOTSURF(SECT)=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTBOTSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section outer radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),2,SECT)
               CURRENTSURFLABEL=0
               DO 3170 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3170          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTORSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTORSURF(SECT)=CURRENTSURFLABEL
               ENDIF
*     Define the GT section inner radius surface.
               CURRENTSURF=GTAXDATA(DESNUM(COLUMN,ROW),1,SECT)
               CURRENTSURFLABEL=0
               DO 3180 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3180          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTSECTIRSURF(SECT)=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTSECTIRSURF(SECT)=CURRENTSURFLABEL
               ENDIF
 3190 CONTINUE
*     Write the GT material cell
        DO 3320 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
```

```
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 3210 RO=1,(ROW-1)
                  DO 3200 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c               (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 3200             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3210          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 3230 RO=ROW,ROW
                     DO 3220 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
 3220                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 3230             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 3250 RO=1,(ROW-1)
```

```
                  DO 3240 CO=1,50
                     IF ((DESNUM(CO,RO).NE.0).AND.
     c                  (BANKNUM(CO,RO).EQ.0)) THEN
                        IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                           EXIT
                        ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                  GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                           EXIT
                        ENDIF
                     ENDIF
 3240             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 3250          CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 3270 RO=1,1
                     DO 3260 CO=1,(COLUMN-1)
                        IF ((DESNUM(CO,RO).NE.0).AND.
     c                     (BANKNUM(CO,RO).EQ.0)) THEN
                           IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTML(CO,RO)
                              EXIT
                           ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.
     c                     GTAXMAT(DESNUM(CO,RO),SECT)) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTAXML(COLUMN,ROW,SECT)=GTAXML(CO,RO,SECT)
                              EXIT
                           ENDIF
                        ENDIF
 3260                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 3270             CONTINUE
               ENDIF
               IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                  GTAXML(COLUMN,ROW,SECT)=MN
*  Check Guide Tube Material
                  IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                     DO 3280 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) GTAXML(COLUMN,ROW,SECT)
```

```
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9301)
                     WRITE(200,7000)
                     WRITE(200,7001)
                     WRITE(200,7002)
                     WRITE(200,9302)
                     WRITE(200,7003)
                     WRITE(200,7004)
                     WRITE(200,7005)
                     WRITE(200,9303)
                     WRITE(200,9304)
                  ENDIF
3280              CONTINUE
                 ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c           .EQ.2) THEN
                    DO 3290 C=1,2
                       IF (C.EQ.1) THEN
                          WRITE(200,9305) GTAXML(COLUMN,ROW,SECT)
                       ELSEIF (C.EQ.2) THEN
                          WRITE(200,9306)
                          WRITE(200,9307)
                          WRITE(200,9308)
                          WRITE(200,9309)
                          WRITE(200,9310)
                          WRITE(200,7006)
                          WRITE(200,7007)
                          WRITE(200,7008)
                          WRITE(200,9311)
                          WRITE(200,9312)
                          WRITE(200,7009)
                          WRITE(200,7010)
                          WRITE(200,7011)
                          WRITE(200,9313)
                          WRITE(200,7012)
                          WRITE(200,7013)
                          WRITE(200,7014)
                          WRITE(200,7015)
                       ENDIF
3290                CONTINUE
                   ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT)
     c             .EQ.3) THEN
                       DO 3300 C=1,2
                          IF (C.EQ.1) THEN
                             WRITE(200,9314) GTAXML(COLUMN,ROW,SECT)
                          ELSEIF (C.EQ.2) THEN
                             WRITE(200,9315)
                             WRITE(200,9316)
                             WRITE(200,9317)
                             WRITE(200,9318)
                             WRITE(200,7016)
                             WRITE(200,7017)
                             WRITE(200,7018)
                             WRITE(200,9319)
                             WRITE(200,9320).
```

```
                              WRITE(200,7019)
                              WRITE(200,7020)
                              WRITE(200,7021)
                              WRITE(200,9321)
                              WRITE(200,7022)
                              WRITE(200,7023)
                              WRITE(200,7024)
                              WRITE(200,7025)
                              WRITE(200,9322)
                              WRITE(200,9323)
                              WRITE(200,9324)
                              WRITE(200,9325)
                              WRITE(200,9326)
                              WRITE(200,9327)
                              WRITE(200,7026)
                              WRITE(200,9328)
                              WRITE(200,9329)
                              WRITE(200,9330)
                           ENDIF
 3300                   CONTINUE
                      ENDIF
                   MN=MN+1
                ENDIF
                IF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.1) THEN
                   CLADRHO=6.56
                ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.2) THEN
                   CLADRHO=7.90
                ELSEIF (GTAXMAT(DESNUM(COLUMN,ROW),SECT).EQ.3) THEN
                   CLADRHO=8.19
                ENDIF
                WRITE(30,3310) LN, GTAXML(COLUMN,ROW,SECT), (-1*CLADRHO),
     c          GTSECTIRSURF(SECT),
     c          (-1*GTSECTORSURF(SECT)), (-1*GTSECTTOPSURF(SECT)),
     c          GTSECTBOTSURF(SECT), BPRAUNIV(COLUMN,ROW)
 3310           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I4,'   $ Guide tube')
                LN=LN+1
 3320        CONTINUE
*     Loop through the spacer and moderator regions along the axial
*     length of the GT (from top to bottom).
            SPACHEIGHT=0.0
            DO 3330 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 3330        CONTINUE
            DO 3520 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                 SPACERTOPSURF=UEFBOTTOMSURF
                 CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c           SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                 CURRENTSURFLABEL=0
                 DO 3340 V=1,(SN-1)
                    IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
```

```
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
3340           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 3350 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
3350           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     C      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     C         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 3360 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
3360           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
```

```
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 3370 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3370           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   WATERREGIONBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
             ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
                SPACERTOPSURF=WATERREGIONBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 3380 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3380           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
             ENDIF
*     Write the current homogenized spacer region cell in this GT universe.
           DO 3510 SECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,3390) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
```

```
3390        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3400) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
3400        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3410) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
3410        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
            WRITE(30,3420) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
3420        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,
     c      '   $ Homogenized region for spacer ',I2)
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c      SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(SPACERTOPSURF))) THEN
            WRITE(30,3430) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c      GTSECTORSURF(SECT),
     c      (-1*SPACERTOPSURF), GTSECTBOTSURF(SECT),
     c      BPRAUNIV(COLUMN,ROW), SPN
```

```
 3430        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF)).AND.
     c       (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(SPACERBOTTOMSURF))) THEN
             WRITE(30,3440) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)),
     c       GTSECTORSURF(SECT),
     c       (-1*GTSECTTOPSURF(SECT)), SPACERBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW), SPN
 3440        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,
     c       '   $ Homogenized region for spacer ',I2)
             LN=LN+1
              ENDIF
*     Write the water region cell below the current homogenized spacer cell
in this GT universe.
              IF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,3450) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 3450        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,3460) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 3460        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
             LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c       (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c       SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
             WRITE(30,3470) LN, BMODML, (-1*MODDENSITY),
     c       GTSECTORSURF(SECT),
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 3470        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I4,'   $ Borated moderator region')
```

```
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).EQ.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,3480) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 3480       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF))) THEN
            WRITE(30,3490) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*WATERREGIONTOPSURF), GTSECTBOTSURF(SECT),
     c      BPRAUNIV(COLUMN,ROW)
 3490       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(SECT)).LT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(SECT)).GT.
     c      SURFVALUESPEC(WATERREGIONBOTTOMSURF))) THEN
            WRITE(30,3500) LN, BMODML, (-1*MODDENSITY),
     c      GTSECTORSURF(SECT),
     c      (-1*GTSECTTOPSURF(SECT)), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 3500       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I4,'   $ Borated moderator region')
            LN=LN+1
             ENDIF
 3510       CONTINUE
 3520    CONTINUE
*  Write the moderator inside of the GT in the BPR universe
        DO 3570 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
           IF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c      SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
*     Write the moderator cells within the GT in this BPR universe.
              WRITE(30,3530) LN, BMODML, (-1*MODDENSITY),
     c        (-1*GTSECTIRSURF(GTSECT)),
     c        BPOCORSURF, (-1*BPCLADTOPSURF),
     c        BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 3530         FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
```

```
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,3540) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPOCORSURF, (-1*BPCLADTOPSURF),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 3540           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LE.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,3550) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          BPCLADBOTTOMSURF,
     c          BPRAUNIV(COLUMN,ROW)
 3550           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).GT.
     c        SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
                WRITE(30,3560) LN, BMODML, (-1*MODDENSITY),
     c          (-1*GTSECTIRSURF(GTSECT)),
     c          BPOCORSURF, (-1*GTSECTTOPSURF(GTSECT)),
     c          GTSECTBOTSURF(GTSECT),
     c          BPRAUNIV(COLUMN,ROW)
 3560           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Borated moderator inside guide tube')
                LN=LN+1
              ENDIF
 3570       CONTINUE
*    Determine the axial GT section which contains the lowest BPR axial
section
           DO 3580 GTSECT=1,NUMOFGTAXS(DESNUM(COLUMN,ROW))
              IF ((SURFVALUESPEC(GTSECTBOTSURF(GTSECT)).LT.
     c      SURFVALUESPEC(BPCLADBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(GTSECT)).GE.
```

```
     c    SURFVALUESPEC(BPCLADBOTTOMSURF))) THEN
             BGT=GTSECT
             EXIT
          ENDIF
 3580  CONTINUE
       DO 3610 GTSECT=BGT,NUMOFGTAXS(DESNUM(COLUMN,ROW))
          IF (GTSECT.EQ.BGT) THEN
             WRITE(30,3590) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*BPCLADBOTTOMSURF),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 3590        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c     . ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ELSE
             WRITE(30,3600) LN, BMODML, (-1*MODDENSITY),
     c       (-1*GTSECTIRSURF(GTSECT)),
     c       (-1*GTSECTTOPSURF(GTSECT)),
     c       GTSECTBOTSURF(GTSECT),
     c       BPRAUNIV(COLUMN,ROW)
 3600        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,
     c       '   $ Borated moderator inside guide tube')
             LN=LN+1
          ENDIF
 3610  CONTINUE
*      Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3620 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 3620          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPLEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*      Write the lower end-fitting cell specification for this BPR universe.
            GTBOTSURF=GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,3630) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
```

```
 3630          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,3640) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW))),
     c         BPRAUNIV(COLUMN,ROW)
 3640          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
               WRITE(30,3650) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)),
     c         (-1*GTSECTBOTSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         (-1*GTSECTORSURF(NUMOFGTAXS(DESNUM(COLUMN,ROW)))),
     c         BPRAUNIV(COLUMN,ROW)
 3650          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
            IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,3660) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 3660          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,3670) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3670          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,3680) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 3680          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c      (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
```

```
     c       (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c       SURFVALUESPEC(UEFBOTTOMSURF))) THEN
                WRITE(30,3690) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1), BPRAUNIV(COLUMN,ROW)
 3690           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3700) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 3700           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3710) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*GTSECTTOPSURF(1)), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 3710           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c        SURFVALUESPEC(GTSECTTOPSURF(1))).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c        SURFVALUESPEC(UEFTOPSURF))) THEN
                WRITE(30,3720) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 3720           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(GTSECTTOPSURF(1))) THEN
                WRITE(30,3730) LN, BMODML,
     c          (-1*MODDENSITY), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTSECTIRSURF(1)),
     c          BPRAUNIV(COLUMN,ROW)
 3730           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,3740) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
```

```
     c          BPRAUNIV(COLUMN,ROW)
 3740           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3750) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3750           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3760) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 3760           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,3770) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 3770           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,3780) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3780           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTSECTTOPSURF(1)).EQ.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,3790) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTSECTORSURF(1),
     c          BPRAUNIV(COLUMN,ROW)
 3790           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
```

```
                LN=LN+1
                WRITE(30,3800) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)),
     c          GTSECTTOPSURF(1),
     c          (-1*UEFTOPSURF), (-1*GTSECTORSURF(1)),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 3800           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
            ENDIF
          ELSE
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
*****************************************************************
          IF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.2) THEN
          DO 3970 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*       Define the upper end-fitting bottom surface.
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c          ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 3810 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                      CURRENTSURFLABEL=V
                      EXIT
                   ENDIF
                ENDIF
 3810           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   UEFBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
```

```
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3820 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3820          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 3830 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3830          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPICIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPICIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 3840 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3840          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
```

```
                  BPICORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPICORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP absorber inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
               CURRENTSURFLABEL=0
               DO 3850 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
                  ENDIF
 3850          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP absorber outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
               CURRENTSURFLABEL=0
               DO 3860 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
                  ENDIF
 3860          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the outer BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
               CURRENTSURFLABEL=0
               DO 3870 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
```

```
                  ENDIF
 3870           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPOCIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPOCIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the outer BPR cladding outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
                CURRENTSURFLABEL=0
                DO 3880 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3880           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPOCORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPOCORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP node bounding surfaces.
              IF (MCNPNODE.EQ.1) THEN
                TOTBPHEIGHT=0.0
                DO 3890 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                   TOTBPHEIGHT=TOTBPHEIGHT+
     c             MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 3890           CONTINUE
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c          TOTBPHEIGHT
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 3900 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3900           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   TOPBPNODETOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
```

```
                  SN=SN+1
                ELSE
                  TOPBPNODETOPSURF=CURRENTSURFLABEL
                ENDIF
                BPNODETOPSURF=TOPBPNODETOPSURF
                CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c          MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 3910 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 3910           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
              ELSEIF (MCNPNODE.NE.1) THEN
                BPNODETOPSURF=BPNODEBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c          MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 3930 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                    CURRENTSURFLABEL=V
                    EXIT
                  ENDIF
                ENDIF
 3930           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
                ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
              ENDIF
            IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
```

```
*     Write the BP node cells in this BPR universe.
         IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
            WRITE(30,3940) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3940       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
        ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,3950) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3950       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,3960) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 3960       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ENDIF
         ENDIF
 3970     CONTINUE
*     Define the BPR cladding top surface.
            CURRENTSURF=TOTBPHEIGHT+
     c      BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c      BPRPLEN(BANKNUM(COLUMN,ROW),1)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 3980 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 3980        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPCLADTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
               SURFVALUESPEC(SN)=CURRENTSURF
               SN=SN+1
```

```
                ELSE
                   BPCLADTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BPR cladding bottom surface.
                CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c          BPRPLEN(BANKNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 3990 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 3990           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPCLADBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPCLADBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
                WRITE(30,4000) LN, (-1*BPIRSURF), BPICORSURF,
     c          (-1*TOPBPNODETOPSURF),
     c          BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4000           FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Burnable poison-to-cladding gap')
                LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
                WRITE(30,4010) LN, (-1*BPOCIRSURF), BPORSURF,
     c          (-1*TOPBPNODETOPSURF),
     c          BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4010           FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Burnable poison-to-cladding gap')
                LN=LN+1
*     Write the annular gap cell in this BPR universe.
                WRITE(30,4020) LN, (-1*BPICIRSURF),
     c          (-1*TOPBPNODETOPSURF),
     c          BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 4020           FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,
     c          '   $ Burnable poison-to-cladding gap')
                LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
                CLADMLUNIQUE=.TRUE.
                LEAVE=.FALSE.
                IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
```

```
                DO 4040 RO=1,(ROW-1)
                   DO 4030 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                         IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                   BPRCLADMAT(BANKNUM(CO,RO))) THEN
                            CLADMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
                      ENDIF
 4030                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4040              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 4060 RO=ROW,ROW
                         DO 4050 CO=1,(COLUMN-1)
                            IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                               IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                         BPRCLADMAT(BANKNUM(CO,RO))) THEN
                                  CLADMLUNIQUE=.FALSE.
                                  LEAVE=.TRUE.
                                  BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                                  EXIT
                               ENDIF
                            ENDIF
                            ENDIF
 4050                       CONTINUE
                            IF (LEAVE.EQ..TRUE.) THEN
                               EXIT
                            ENDIF
 4060                    CONTINUE
                      ENDIF
                   ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                      DO 4080 RO=1,(ROW-1)
                         DO 4070 CO=1,50
                            IF (BANKNUM(CO,RO).NE.0) THEN
                            IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                               IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                         BPRCLADMAT(BANKNUM(CO,RO))) THEN
                                  CLADMLUNIQUE=.FALSE.
                                  LEAVE=.TRUE.
                                  BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                                  EXIT
                               ENDIF
                            ENDIF
                            ENDIF
 4070                       CONTINUE
                            IF (LEAVE.EQ..TRUE.) THEN
```

```
                  EXIT
               ENDIF
 4080        CONTINUE
           ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
              DO 4100 RO=1,1
                 DO 4090 CO=1,(COLUMN-1)
                    IF (BANKNUM(CO,RO).NE.0) THEN
                    IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                       IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                 BPRCLADMAT(BANKNUM(CO,RO))) THEN
                          CLADMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
                    ENDIF
 4090               CONTINUE
                    IF (LEAVE.EQ..TRUE.) THEN
                       EXIT
                    ENDIF
 4100            CONTINUE
              ENDIF
              IF (CLADMLUNIQUE.EQ..TRUE.) THEN
                 BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                    IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                       DO 4110 C=1,2
                          IF (C.EQ.1) THEN
                             WRITE(200,9300) BPCLADML(COLUMN,ROW)
                          ELSEIF (C.EQ.2) THEN
                             WRITE(200,9301)
                             WRITE(200,7000)
                             WRITE(200,7001)
                             WRITE(200,7002)
                             WRITE(200,9302)
                             WRITE(200,7003)
                             WRITE(200,7004)
                             WRITE(200,7005)
                             WRITE(200,9303)
                             WRITE(200,9304)
                          ENDIF
 4110                  CONTINUE
                    ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c              .EQ.2) THEN
                       DO 4120 C=1,2
                          IF (C.EQ.1) THEN
                             WRITE(200,9305) BPCLADML(COLUMN,ROW)
                          ELSEIF (C.EQ.2) THEN
                             WRITE(200,9306)
                             WRITE(200,9307)
                             WRITE(200,9308)
                             WRITE(200,9309)
                             WRITE(200,9310)
```

```
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
4120                CONTINUE
                  ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     C            .EQ.3) THEN
                     DO 4130 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
                           WRITE(200,9317)
                           WRITE(200,9318)
                           WRITE(200,7016)
                           WRITE(200,7017)
                           WRITE(200,7018)
                           WRITE(200,9319)
                           WRITE(200,9320)
                           WRITE(200,7019)
                           WRITE(200,7020)
                           WRITE(200,7021)
                           WRITE(200,9321)
                           WRITE(200,7022)
                           WRITE(200,7023)
                           WRITE(200,7024)
                           WRITE(200,7025)
                           WRITE(200,9322)
                           WRITE(200,9323)
                           WRITE(200,9324)
                           WRITE(200,9325)
                           WRITE(200,9326)
                           WRITE(200,9327)
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
4130                 CONTINUE
                  ENDIF
               MN=MN+1
            ENDIF
            IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
```

```
                CLADRHO=6.56
              ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                CLADRHO=7.90
              ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                CLADRHO=8.19
              ENDIF
              WRITE(30,4140) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c        BPOCIRSURF,
     c        (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 4140        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ BPR cladding')
              LN=LN+1
              WRITE(30,4150) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c        BPICIRSURF,
     c        (-1*BPICORSURF), (-1*TOPBPNODETOPSURF), BPNODEBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 4150        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ BPR cladding')
              LN=LN+1
*      Write the BPR upper plenum cell in this BPR universe.
*      Determine if the BPR upper plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the upper plenum material specification label.
              BPRUPMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                 DO 4170 RO=1,(ROW-1)
                    DO 4160 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             BPRUPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 4160                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4170              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 4190 RO=ROW,ROW
                         DO 4180 CO=1,(COLUMN-1)
                            IF (BANKNUM(CO,RO).NE.0) THEN
                               IF (BANKNUM(COLUMN,ROW).EQ.
     c                            BANKNUM(CO,RO)) THEN
                                  BPRUPMLUNIQUE=.FALSE.
                                  LEAVE=.TRUE.
                                  BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                                  EXIT
                               ENDIF
```

```
                        ENDIF
 4180                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4190             CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 4210 RO=1,(ROW-1)
                   DO 4200 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            BPRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 4200                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4210             CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 4230 RO=1,1
                   DO 4220 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            BPRUPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 4220                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4230             CONTINUE
               ENDIF
          IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c    SURFVALUESPEC(UEFTOPSURF)) THEN
             IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
                BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                DO 4260 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,4240) BPRUPML(COLUMN,ROW),
     c                BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4240                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '   $ Burnable Poison Rod Upper Plenum')
```

```
                  ELSE
                     WRITE(200,4250)
     c               BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c               (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 4250                FORMAT(T9,A9,3X,G14.6)
                  ENDIF
 4260           CONTINUE
              MN=MN+1
           ENDIF
           WRITE(30,4270) LN, BPRUPML(COLUMN,ROW),
     c     (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c     TOPBPNODETOPSURF,
     c     (-1*BPCLADTOPSURF), (-1*BPOCIRSURF),
     c     BPRAUNIV(COLUMN,ROW)
 4270      FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c     ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
           LN=LN+1
        ENDIF
*     Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*     Determine if the BPR lower plenum material specification has
*     previously been defined.  If it has been previously defined, determine
*     the lower plenum material specification label.
           BPRLPMLUNIQUE=.TRUE.
           LEAVE=.FALSE.
           IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
              DO 4290 RO=1,(ROW-1)
                 DO 4280 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                       IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                          BPRLPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
                          BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                          EXIT
                       ENDIF
                    ENDIF
 4280            CONTINUE
                 IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                 ENDIF
 4290         CONTINUE
              IF (LEAVE.EQ..FALSE.) THEN
                 DO 4310 RO=ROW,ROW
                    DO 4300 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                    BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 4300                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
```

```
                        EXIT
                     ENDIF
 4310             CONTINUE
                ENDIF
             ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                DO 4330 RO=1,(ROW-1)
                   DO 4320 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            BPRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 4320                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 4330              CONTINUE
             ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                DO 4350 RO=1,1
                   DO 4340 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.
     c                      BANKNUM(CO,RO)) THEN
                            BPRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 4340                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 4350              CONTINUE
             ENDIF
             IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
                BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                DO 4380 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                   IF (C.EQ.1) THEN
                      WRITE(200,4360) BPRLPML(COLUMN,ROW),
     c                BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 4360                 FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                '   $ Burnable Poison Rod Lower Plenum')
                   ELSE
                      WRITE(200,4370)
     c                BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 4370                 FORMAT(T9,A9,3X,G14.6)
```

```
                  ENDIF
 4380            CONTINUE
               MN=MN+1
             ENDIF
             WRITE(30,4390) LN, BPRLPML(COLUMN,ROW),
     c       (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c       (-1*BPNODEBOTTOMSURF), (-1*BPOCIRSURF),
     c       BPRAUNIV(COLUMN,ROW)
 4390        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
             LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
              DO 4430 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                IF (REGION.EQ.1) THEN
                   REGIONTOPSURF=SYSTEMTOP
                   CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c             REGABOVEBPRA(REGION,1)
                ENDIF
                CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c          REGABOVEBPRA(REGION,1)
                IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                   REGIONBOTTOMSURF=UEFTOPSURF
                ELSE
                CURRENTSURFLABEL=0
                DO 4400 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4400            CONTINUE
                 IF (CURRENTSURFLABEL.EQ.0) THEN
                    REGIONBOTTOMSURF=SN
                    SURFTYPESPEC(SN)='PZ'
                    SURFVALUESPEC(SN)=CURRENTSURF
                    SN=SN+1
                 ELSE
                    REGIONBOTTOMSURF=CURRENTSURFLABEL
                 ENDIF
                 ENDIF
*     Write the cell specification for the BPR universe upper region.
              IF (REGION.EQ.1) THEN
                 WRITE(30,4410) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c           (-1*REGABOVEBPRA(REGION,2)),
     c           REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4410            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c           ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                 LN=LN+1
                 REGIONTOPSURF=REGIONBOTTOMSURF
              ELSE
```

```
                  WRITE(30,4420) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4420             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
                ENDIF
 4430           CONTINUE
*     Define the GT top surface.
                CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
                IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                   CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
                ENDIF
                CURRENTSURFLABEL=0
                DO 4440 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4440           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTTOPSURF=CURRENTSURFLABEL
                ENDIF
*     Define the GT bottom surface.
                CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
                CURRENTSURFLABEL=0
                DO 4450 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4450           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTBOTSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTBOTSURF=CURRENTSURFLABEL
                ENDIF
*     Define the GT outer radius surface.
                CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 4460 V=1,(SN-1)
```

```
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4460          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT inner radius surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 4470 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4470          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the lower end-fitting top surface.
               CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 4480 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
               IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4480          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPLEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
```

```
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,4490) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
 4490          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Assembly lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,4500) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTORSURF, BPRAUNIV(COLUMN,ROW)
 4500          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
               LN=LN+1
               WRITE(30,4510) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c         (-1*GTORSURF), BPRAUNIV(COLUMN,ROW)
 4510          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
            IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,4520) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 4520          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,4530) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4530          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,4540) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 4540          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTTOPSURF)).AND.
```

```
     c       (SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,4550) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4550          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,4560) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c         (-1*UEFTOPSURF), (-1*GTORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 4560          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,4570) LN, BMODML,
     c         (-1*MODDENSITY), BPCLADTOPSURF,
     c         (-1*GTTOPSURF), (-1*GTIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 4570          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(GTTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).EQ.
     c      SURFVALUESPEC(UEFTOPSURF))) THEN
               WRITE(30,4580) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4580          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(GTTOPSURF)) THEN
               WRITE(30,4590) LN, BMODML,
     c         (-1*MODDENSITY), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*GTIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 4590          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ENDIF
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LT.
     c      SURFVALUESPEC(BPCLADTOPSURF))) THEN
              WRITE(30,4600) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
```

```
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4600           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4610) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4610           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4620) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPOCORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 4620           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,4630) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4630           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4640) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4640           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).EQ.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,4650) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 4650           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,4660) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
```

```
     c          BPOCORSURF, BPRAUNIV(COLUMN,ROW)
 4660           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
*     Write the GT material cell in this BPR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 4680 RO=1,(ROW-1)
                  DO 4670 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 4670             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 4680          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 4700 RO=ROW,ROW
                     DO 4690 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 4690                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 4700             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 4720 RO=1,(ROW-1)
                  DO 4710 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
```

```
                        LEAVE=.TRUE.
                        GTML(COLUMN,ROW)=GTML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
4710           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
4720        CONTINUE
         ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
            DO 4740 RO=1,1
               DO 4730 CO=1,(COLUMN-1)
                  IF (DESNUM(CO,RO).NE.0) THEN
                     IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c               GTMAT(DESNUM(CO,RO))) THEN
                        CLADMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        GTML(COLUMN,ROW)=GTML(CO,RO)
                        EXIT
                     ENDIF
                  ENDIF
4730           CONTINUE
               IF (LEAVE.EQ..TRUE.) THEN
                  EXIT
               ENDIF
4740        CONTINUE
         ENDIF
         IF (CLADMLUNIQUE.EQ..TRUE.) THEN
            GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
            IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
               DO 4750 C=1,2
                  IF (C.EQ.1) THEN
                     WRITE(200,9300) GTML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9301)
                     WRITE(200,7000)
                     WRITE(200,7001)
                     WRITE(200,7002)
                     WRITE(200,9302)
                     WRITE(200,7003)
                     WRITE(200,7004)
                     WRITE(200,7005)
                     WRITE(200,9303)
                     WRITE(200,9304)
                  ENDIF
4750           CONTINUE
            ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c      .EQ.2) THEN
               DO 4760 C=1,2
                  IF (C.EQ.1) THEN
                     WRITE(200,9305) GTML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
```

```
                        WRITE(200,9306)
                        WRITE(200,9307)
                        WRITE(200,9308)
                        WRITE(200,9309)
                        WRITE(200,9310)
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
4760                 CONTINUE
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c            .EQ.3) THEN
                     DO 4770 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) GTML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
                           WRITE(200,9317)
                           WRITE(200,9318)
                           WRITE(200,7016)
                           WRITE(200,7017)
                           WRITE(200,7018)
                           WRITE(200,9319)
                           WRITE(200,9320)
                           WRITE(200,7019)
                           WRITE(200,7020)
                           WRITE(200,7021)
                           WRITE(200,9321)
                           WRITE(200,7022)
                           WRITE(200,7023)
                           WRITE(200,7024)
                           WRITE(200,7025)
                           WRITE(200,9322)
                           WRITE(200,9323)
                           WRITE(200,9324)
                           WRITE(200,9325)
                           WRITE(200,9326)
                           WRITE(200,9327)
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
```

```
 4770                 CONTINUE
                     ENDIF
                   MN=MN+1
                 ENDIF
                 IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                   CLADRHO=6.56
                 ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
                   CLADRHO=7.90
                 ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                   CLADRHO=8.19
                 ENDIF
                 WRITE(30,4780) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c           GTIRSURF,
     c           (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c           BPRAUNIV(COLUMN,ROW)
 4780            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Guide tube region')
                 LN=LN+1
*      Write the moderator cells within the GT in this BPR universe.
                 IF (SURFVALUESPEC(BPCLADTOPSURF).GE.
     c           SURFVALUESPEC(GTTOPSURF)) THEN
                   WRITE(30,4790) LN, BMODML, (-1*MODDENSITY),
     c             (-1*GTIRSURF),
     c             BPOCORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c             BPRAUNIV(COLUMN,ROW)
 4790              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c             ' IMP:N=1 U=',I3,
     c             '   $ Borated moderator inside guide tube')
                   LN=LN+1
                 ELSEIF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c           SURFVALUESPEC(GTTOPSURF)) THEN
                   WRITE(30,4800) LN, BMODML, (-1*MODDENSITY),
     c             (-1*GTIRSURF),
     c             BPOCORSURF, (-1*BPCLADTOPSURF), GTBOTSURF,
     c             BPRAUNIV(COLUMN,ROW)
 4800              FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c             ' IMP:N=1 U=',I3,
     c             '   $ Borated moderator inside guide tube')
                   LN=LN+1
                 ENDIF
                 WRITE(30,4810) LN, BMODML, (-1*MODDENSITY),
     c           (-1*BPOCORSURF), (-1*BPCLADBOTTOMSURF), GTBOTSURF,
     c           BPRAUNIV(COLUMN,ROW)
 4810            FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c           ' IMP:N=1 U=',I3,'   $ Borated moderator inside guide tube')
                 LN=LN+1
*      Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*      Define the upper region lower surface.
                  DO 4850 REGION=1,NUMREGABOVEBPRA
*      Determine the current upper region's lower surface specification.
                    IF (REGION.EQ.1) THEN
                      REGIONTOPSURF=SYSTEMTOP
                      CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
```

```
     c               REGABOVEBPRA(REGION,1)
                    ENDIF
                    CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c              REGABOVEBPRA(REGION,1)
                    IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                       REGIONBOTTOMSURF=UEFTOPSURF
                    ELSE
                    CURRENTSURFLABEL=0
                    DO 4820 V=1,(SN-1)
                       IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                             CURRENTSURFLABEL=V
                             EXIT
                          ENDIF
                       ENDIF
 4820                 CONTINUE
                    IF (CURRENTSURFLABEL.EQ.0) THEN
                       REGIONBOTTOMSURF=SN
                       SURFTYPESPEC(SN)='PZ'
                       SURFVALUESPEC(SN)=CURRENTSURF
                       SN=SN+1
                    ELSE
                       REGIONBOTTOMSURF=CURRENTSURFLABEL
                    ENDIF
                    ENDIF
*     Write the cell specification for the BPR universe upper region.
                 IF (REGION.EQ.1) THEN
                    WRITE(30,4830) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c              (-1*REGABOVEBPRA(REGION,2)),
     c              REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4830               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c              ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ELSE
                    WRITE(30,4840) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c              (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c              REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 4840               FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c              ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                    LN=LN+1
                    REGIONTOPSURF=REGIONBOTTOMSURF
                 ENDIF
 4850            CONTINUE
            SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the BPR (from top to bottom).
            DO 4860 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 4860       CONTINUE
            DO 4940 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
*     Define the homogenized spacer region bounding surfaces.
              IF (SPN.EQ.1) THEN
                 SPACERTOPSURF=UEFBOTTOMSURF
```

```
                CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 4870 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4870           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   SPACERBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
                WATERREGIONTOPSURF=SPACERBOTTOMSURF
                CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
                CURRENTSURFLABEL=0
                DO 4880 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4880           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   WATERREGIONBOTTOMSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
                ENDIF
             ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c       NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
                SPACERTOPSURF=WATERREGIONBOTTOMSURF
                CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c          SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
                CURRENTSURFLABEL=0
                DO 4890 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4890           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   SPACERBOTTOMSURF=SN
```

```
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 4900 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
4900           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 4910 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
4910           CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this BPR universe.
            WRITE(30,4920) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
     c      (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c      (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c      SPN
```

```
 4920       FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
            LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this BPR universe.
            WRITE(30,4930) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c      (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW)
 4930       FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c      ' IMP:N=1 U=',I3,'   $ Borated moderator')
            LN=LN+1
 4940     CONTINUE
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
**********************************************************************
          ELSEIF (WBPRA(BANKNUM(COLUMN,ROW)).EQ.3) THEN
          DO 5110 MCNPNODE=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
*     Define the upper end-fitting bottom surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 4950 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 4950          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
```

```
                  SN=SN+1
               ELSE
                  UEFBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Define the upper end-fitting top surface.
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),1)+
     c         ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 4960 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4960          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  UEFTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  UEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding inner radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),1)
               CURRENTSURFLABEL=0
               DO 4970 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 4970          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPICIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPICIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the inner BPR cladding outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 4980 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
```

```
 4980           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPICORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPICORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP absorber inner radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),3)
                CURRENTSURFLABEL=0
                DO 4990 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 4990           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the BP absorber outer radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),4)
                CURRENTSURFLABEL=0
                DO 5000 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5000           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   BPORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the outer BPR cladding inner radius.
                CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),5)
                CURRENTSURFLABEL=0
                DO 5010 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
```

```
                        EXIT
                     ENDIF
                  ENDIF
 5010          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPOCIRSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPOCIRSURF=CURRENTSURFLABEL
               ENDIF
*     Define the outer BPR cladding outer radius.
               CURRENTSURF=BPRAXDIM(BANKNUM(COLUMN,ROW),6)
               CURRENTSURFLABEL=0
               DO 5020 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'CZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5020          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPOCORSURF=SN
                  SURFTYPESPEC(SN)='CZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPOCORSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BP node bounding surfaces.
            IF (MCNPNODE.EQ.1) THEN
               TOTBPHEIGHT=0.0
               DO 5030 Z=1,NUMOFBPRANODES(BANKNUM(COLUMN,ROW))
                  TOTBPHEIGHT=TOTBPHEIGHT+
     c            MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),Z)
 5030          CONTINUE
               CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c         TOTBPHEIGHT
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 5040 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5040          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  TOPBPNODETOPSURF=SN
```

```
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  TOPBPNODETOPSURF=CURRENTSURFLABEL
               ENDIF
               BPNODETOPSURF=TOPBPNODETOPSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 5050 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 5050          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (MCNPNODE.NE.1) THEN
               BPNODETOPSURF=BPNODEBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(BPNODETOPSURF)-
     c         MCNPBPRAHEIGHT(BANKNUM(COLUMN,ROW),MCNPNODE)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 5070 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 5070          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPNODEBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPNODEBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ENDIF
```

```
         IF (SURFVALUESPEC(BPNODEBOTTOMSURF).LT.
     c   SURFVALUESPEC(UEFTOPSURF)) THEN
*     Write the BP node cells in this BPR universe.
         IF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).EQ.1)) THEN
            WRITE(30,5080) LN, BPNODEML,
     c      (-1*AL2O3DENSITY(BANKNUM(COLUMN,ROW))), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 5080       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSEIF ((BPRABSNODE(BANKNUM(COLUMN,ROW),MCNPNODE).EQ.'Y').AND.
     c      (BPNONABSMAT(BANKNUM(COLUMN,ROW)).NE.1)) THEN
            WRITE(30,5090) LN, BPNODEML,
     c      (-1*NONBPMATDATA(BANKNUM(COLUMN,ROW),1)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 5090       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ELSE
            WRITE(30,5100) LN, BPNODEML,
     c      (-1*BPDENTOGO(COLUMN,ROW,MCNPNODE)), (-1*BPORSURF),
     c      BPIRSURF, (-1*BPNODETOPSURF), BPNODEBOTTOMSURF,
     c      BPRAUNIV(COLUMN,ROW), MCNPNODE
 5100       FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c      1X,I4,' IMP:N=1 U=',I3,'   $ Burnable poison node ',I2)
            LN=LN+1
            BPNODEML=BPNODEML+1
         ENDIF
         ENDIF
 5110     CONTINUE
*     Define the BPR cladding top surface.
            CURRENTSURF=TOTBPHEIGHT+
     c      BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))+
     c      BPRPLEN(BANKNUM(COLUMN,ROW),1)
            IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
               CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
            ENDIF
            CURRENTSURFLABEL=0
            DO 5120 V=1,(SN-1)
               IF (SURFTYPESPEC(V).EQ.'PZ') THEN
          IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5120        CONTINUE
            IF (CURRENTSURFLABEL.EQ.0) THEN
               BPCLADTOPSURF=SN
               SURFTYPESPEC(SN)='PZ'
```

```
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the BPR cladding bottom surface.
               CURRENTSURF=BOTBPNODEHEIGHT(BANKNUM(COLUMN,ROW))-
     c         BPRPLEN(BANKNUM(COLUMN,ROW),2)
               CURRENTSURFLABEL=0
               DO 5130 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5130          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  BPCLADBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  BPCLADBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
*     Write the inner BP-to-cladding gap cell in this BPR universe.
               WRITE(30,5140) LN, (-1*BPIRSURF), BPICORSURF,
     c         (-1*TOPBPNODETOPSURF),
     c         BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5140          FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Burnable poison-to-cladding gap')
               LN=LN+1
*     Write the outer BP-to-cladding gap cell in this BPR universe.
               WRITE(30,5150) LN, (-1*BPOCIRSURF), BPORSURF,
     c         (-1*TOPBPNODETOPSURF),
     c         BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5150          FORMAT(T1,I4,T6,'0',T25,I4,1X,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Burnable poison-to-cladding gap')
               LN=LN+1
*     Write the annular water cell in this BPR universe.
               WRITE(30,5160) LN, BMODML,
     c         (-1*MODDENSITY), (-1*BPICIRSURF),
     c         (-1*TOPBPNODETOPSURF),
     c         BPNODEBOTTOMSURF, BPRAUNIV(COLUMN,ROW)
 5160          FORMAT(T1,I4,T6,I4,T11,G14.6,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,
     c         '   $ Burnable poison-to-cladding gap')
               LN=LN+1
*     Write the BPR cladding cell in this BPR universe.
*     Determine if the BPR cladding material specification has
*     previously been defined.  If it has been previously defined, determine
*     the cladding material specification label.
```

```
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 5180 RO=1,(ROW-1)
                  DO 5170 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
 5170             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5180          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 5200 RO=ROW,ROW
                     DO 5190 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                           IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                     BPRCLADMAT(BANKNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
                        ENDIF
 5190                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5200             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 5220 RO=1,(ROW-1)
                  DO 5210 CO=1,50
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).EQ.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
```

```
                  ENDIF
5210              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
5220           CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 5240 RO=1,1
                  DO 5230 CO=1,(COLUMN-1)
                     IF (BANKNUM(CO,RO).NE.0) THEN
                     IF (BANKDES(BANKNUM(CO,RO)).NE.'BPRA ') THEN
                        IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.
     c                  BPRCLADMAT(BANKNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           BPCLADML(COLUMN,ROW)=BPCLADML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
                     ENDIF
5230              CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
5240           CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               BPCLADML(COLUMN,ROW)=MN
*  Check BPR Cladding Material
                  IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                     DO 5250 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9300) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9301)
                           WRITE(200,7000)
                           WRITE(200,7001)
                           WRITE(200,7002)
                           WRITE(200,9302)
                           WRITE(200,7003)
                           WRITE(200,7004)
                           WRITE(200,7005)
                           WRITE(200,9303)
                           WRITE(200,9304)
                        ENDIF
5250                 CONTINUE
                  ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     c            .EQ.2) THEN
                     DO 5260 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9305) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9306)
                           WRITE(200,9307)
```

```
                        WRITE(200,9308)
                        WRITE(200,9309)
                        WRITE(200,9310)
                        WRITE(200,7006)
                        WRITE(200,7007)
                        WRITE(200,7008)
                        WRITE(200,9311)
                        WRITE(200,9312)
                        WRITE(200,7009)
                        WRITE(200,7010)
                        WRITE(200,7011)
                        WRITE(200,9313)
                        WRITE(200,7012)
                        WRITE(200,7013)
                        WRITE(200,7014)
                        WRITE(200,7015)
                     ENDIF
5260                CONTINUE
                  ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW))
     C            .EQ.3) THEN
                     DO 5270 C=1,2
                        IF (C.EQ.1) THEN
                           WRITE(200,9314) BPCLADML(COLUMN,ROW)
                        ELSEIF (C.EQ.2) THEN
                           WRITE(200,9315)
                           WRITE(200,9316)
                           WRITE(200,9317)
                           WRITE(200,9318)
                           WRITE(200,7016)
                           WRITE(200,7017)
                           WRITE(200,7018)
                           WRITE(200,9319)
                           WRITE(200,9320)
                           WRITE(200,7019)
                           WRITE(200,7020)
                           WRITE(200,7021)
                           WRITE(200,9321)
                           WRITE(200,7022)
                           WRITE(200,7023)
                           WRITE(200,7024)
                           WRITE(200,7025)
                           WRITE(200,9322)
                           WRITE(200,9323)
                           WRITE(200,9324)
                           WRITE(200,9325)
                           WRITE(200,9326)
                           WRITE(200,9327)
                           WRITE(200,7026)
                           WRITE(200,9328)
                           WRITE(200,9329)
                           WRITE(200,9330)
                        ENDIF
5270                 CONTINUE
                  ENDIF
```

```
                MN=MN+1
              ENDIF
              IF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.1) THEN
                CLADRHO=6.56
              ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.2) THEN
                CLADRHO=7.90
              ELSEIF (BPRCLADMAT(BANKNUM(COLUMN,ROW)).EQ.3) THEN
                CLADRHO=8.19
              ENDIF
              WRITE(30,5280) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c        BPOCIRSURF,
     c        (-1*BPOCORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 5280        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ BPR cladding')
              LN=LN+1
              WRITE(30,5290) LN, BPCLADML(COLUMN,ROW), (-1*CLADRHO),
     c        BPICIRSURF,
     c        (-1*BPICORSURF), (-1*BPCLADTOPSURF), BPCLADBOTTOMSURF,
     c        BPRAUNIV(COLUMN,ROW)
 5290        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ BPR cladding')
              LN=LN+1
*      Write the BPR upper plenum cell in this BPR universe.
*      Determine if the BPR upper plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the upper plenum material specification label.
              BPRUPMLUNIQUE=.TRUE.
              LEAVE=.FALSE.
              IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 5310 RO=1,(ROW-1)
                  DO 5300 CO=1,50
                    IF (BANKNUM(CO,RO).NE.0) THEN
                      IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                        BPRUPMLUNIQUE=.FALSE.
                        LEAVE=.TRUE.
                        BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                        EXIT
                      ENDIF
                    ENDIF
 5300             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                    EXIT
                  ENDIF
 5310           CONTINUE
                IF (LEAVE.EQ..FALSE.) THEN
                  DO 5330 RO=ROW,ROW
                    DO 5320 CO=1,(COLUMN-1)
                      IF (BANKNUM(CO,RO).NE.0) THEN
                        IF (BANKNUM(COLUMN,ROW).EQ.
     c                  BANKNUM(CO,RO)) THEN
                          BPRUPMLUNIQUE=.FALSE.
                          LEAVE=.TRUE.
```

```
                                BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                                EXIT
                             ENDIF
                          ENDIF
 5320                  CONTINUE
                       IF (LEAVE.EQ..TRUE.) THEN
                          EXIT
                       ENDIF
 5330                CONTINUE
                  ENDIF
               ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                  DO 5350 RO=1,(ROW-1)
                     DO 5340 CO=1,50
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 5340                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5350             CONTINUE
               ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                  DO 5370 RO=1,1
                     DO 5360 CO=1,(COLUMN-1)
                        IF (BANKNUM(CO,RO).NE.0) THEN
                           IF (BANKNUM(COLUMN,ROW).EQ.
     c                        BANKNUM(CO,RO)) THEN
                              BPRUPMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              BPRUPML(COLUMN,ROW)=BPRUPML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 5360                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5370             CONTINUE
               ENDIF
            IF (SURFVALUESPEC(TOPBPNODETOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)) THEN
               IF (BPRUPMLUNIQUE.EQ..TRUE.) THEN
                  BPRUPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Upper Plenum Regions
                  DO 5400 C=1,BPRUPLENMAT(BANKNUM(COLUMN,ROW),2)
                     IF (C.EQ.1) THEN
                        WRITE(200,5380) BPRUPML(COLUMN,ROW),
     c                  BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
```

```
     c                  (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5380                   FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                  '   $ Burnable Poison Rod Upper Plenum')
                      ELSE
                         WRITE(200,5390)
     c                   BPRUPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                   (-1*BPRUPLENWTS(BANKNUM(COLUMN,ROW),C))
 5390                    FORMAT(T9,A9,3X,G14.6)
                      ENDIF
 5400              CONTINUE
                MN=MN+1
             ENDIF
             WRITE(30,5410) LN, BPRUPML(COLUMN,ROW),
     c       (-1*BPRUPLENMAT(BANKNUM(COLUMN,ROW),1)),
     c       TOPBPNODETOPSURF,
     c       (-1*BPCLADTOPSURF), (-1*BPCLADIRSURF),
     c       BPRAUNIV(COLUMN,ROW)
 5410        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ BPR upper plenum region')
             LN=LN+1
          ENDIF
*      Write the BPR lower plenum cell (lower end plug) in this BPR universe.
*      Determine if the BPR lower plenum material specification has
*      previously been defined.  If it has been previously defined, determine
*      the lower plenum material specification label.
             BPRLPMLUNIQUE=.TRUE.
             LEAVE=.FALSE.
             IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
                DO 5430 RO=1,(ROW-1)
                   DO 5420 CO=1,50
                      IF (BANKNUM(CO,RO).NE.0) THEN
                         IF (BANKNUM(COLUMN,ROW).EQ.BANKNUM(CO,RO)) THEN
                            BPRLPMLUNIQUE=.FALSE.
                            LEAVE=.TRUE.
                            BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                            EXIT
                         ENDIF
                      ENDIF
 5420                 CONTINUE
                      IF (LEAVE.EQ..TRUE.) THEN
                         EXIT
                      ENDIF
 5430              CONTINUE
                   IF (LEAVE.EQ..FALSE.) THEN
                      DO 5450 RO=ROW,ROW
                         DO 5440 CO=1,(COLUMN-1)
                            IF (BANKNUM(CO,RO).NE.0) THEN
                               IF (BANKNUM(COLUMN,ROW).EQ.
     c                         BANKNUM(CO,RO)) THEN
                                  BPRLPMLUNIQUE=.FALSE.
                                  LEAVE=.TRUE.
                                  BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                                  EXIT
                               ENDIF
```

```
                        ENDIF
 5440                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5450             CONTINUE
                ENDIF
              ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
                 DO 5470 RO=1,(ROW-1)
                    DO 5460 CO=1,50
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 5460                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5470             CONTINUE
              ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
                 DO 5490 RO=1,1
                    DO 5480 CO=1,(COLUMN-1)
                       IF (BANKNUM(CO,RO).NE.0) THEN
                          IF (BANKNUM(COLUMN,ROW).EQ.
     c                       BANKNUM(CO,RO)) THEN
                             BPRLPMLUNIQUE=.FALSE.
                             LEAVE=.TRUE.
                             BPRLPML(COLUMN,ROW)=BPRLPML(CO,RO)
                             EXIT
                          ENDIF
                       ENDIF
 5480                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5490             CONTINUE
              ENDIF
              IF (BPRLPMLUNIQUE.EQ..TRUE.) THEN
                 BPRLPML(COLUMN,ROW)=MN
*  Check Burnable Poison Rod Lower Plenum Regions
                 DO 5520 C=1,BPRLPLENMAT(BANKNUM(COLUMN,ROW),2)
                    IF (C.EQ.1) THEN
                       WRITE(200,5500) BPRLPML(COLUMN,ROW),
     c                 BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                 (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5500                  FORMAT(T1,'M',I4,T9,A9,3X,G14.6,
     c                 '   $ Burnable Poison Rod Lower Plenum')
                    ELSE
                       WRITE(200,5510)
```

```
     c                    BPRLPLENZAIDS(BANKNUM(COLUMN,ROW),C),
     c                    (-1*BPRLPLENWTS(BANKNUM(COLUMN,ROW),C))
 5510                     FORMAT(T9,A9,3X,G14.6)
                       ENDIF
 5520                CONTINUE
                  MN=MN+1
               ENDIF
               WRITE(30,5530) LN, BPRLPML(COLUMN,ROW),
     c         (-1*BPRLPLENMAT(BANKNUM(COLUMN,ROW),1)), BPCLADBOTTOMSURF,
     c         (-1*BPNODEBOTTOMSURF), (-1*BPCLADIRSURF),
     c         BPRAUNIV(COLUMN,ROW)
 5530        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ BPR lower plenum region')
               LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                  DO 5570 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
                     IF (REGION.EQ.1) THEN
                        REGIONTOPSURF=SYSTEMTOP
                        CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c                  REGABOVEBPRA(REGION,1)
                     ENDIF
                     CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c               REGABOVEBPRA(REGION,1)
                     IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                        REGIONBOTTOMSURF=UEFTOPSURF
                     ELSE
                     CURRENTSURFLABEL=0
                     DO 5540 V=1,(SN-1)
                        IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                  IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                              CURRENTSURFLABEL=V
                              EXIT
                           ENDIF
                        ENDIF
 5540                CONTINUE
                     IF (CURRENTSURFLABEL.EQ.0) THEN
                        REGIONBOTTOMSURF=SN
                        SURFTYPESPEC(SN)='PZ'
                        SURFVALUESPEC(SN)=CURRENTSURF
                        SN=SN+1
                     ELSE
                        REGIONBOTTOMSURF=CURRENTSURFLABEL
                     ENDIF
                     ENDIF
*     Write the cell specification for the BPR universe upper region.
                  IF (REGION.EQ.1) THEN
                     WRITE(30,5550) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c               (-1*REGABOVEBPRA(REGION,2)),
     c               REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5550                FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c               ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
```

```
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,5560) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5560             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 5570          CONTINUE
*     Define the GT top surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),3)
               IF (CURRENTSURF.GE.SURFVALUESPEC(UEFTOPSURF)) THEN
                  CURRENTSURF=SURFVALUESPEC(UEFTOPSURF)
               ENDIF
               CURRENTSURFLABEL=0
               DO 5580 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5580          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTTOPSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT bottom surface.
               CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),4)
               CURRENTSURFLABEL=0
               DO 5590 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                     CURRENTSURFLABEL=V
                     EXIT
                  ENDIF
               ENDIF
 5590          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  GTBOTSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  GTBOTSURF=CURRENTSURFLABEL
               ENDIF
*     Define the GT outer radius surface.
```

```
                CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 5600 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5600           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTORSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTORSURF=CURRENTSURFLABEL
                ENDIF
*     Define the GT inner radius surface.
                CURRENTSURF=GTDATA(DESNUM(COLUMN,ROW),1)
                CURRENTSURFLABEL=0
                DO 5610 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'CZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5610           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   GTIRSURF=SN
                   SURFTYPESPEC(SN)='CZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
                   GTIRSURF=CURRENTSURFLABEL
                ENDIF
*     Define the lower end-fitting top surface.
                CURRENTSURF=ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)
                CURRENTSURFLABEL=0
                DO 5620 V=1,(SN-1)
                   IF (SURFTYPESPEC(V).EQ.'PZ') THEN
              IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                         CURRENTSURFLABEL=V
                         EXIT
                      ENDIF
                   ENDIF
 5620           CONTINUE
                IF (CURRENTSURFLABEL.EQ.0) THEN
                   BPLEFTOPSURF=SN
                   SURFTYPESPEC(SN)='PZ'
                   SURFVALUESPEC(SN)=CURRENTSURF
                   SN=SN+1
                ELSE
```

```
                  BPLEFTOPSURF=CURRENTSURFLABEL
               ENDIF
*     Write the lower end-fitting cell specification for this BPR universe.
            IF (SURFVALUESPEC(GTBOTSURF).GE.
     c      ENDFITHEIGHT(DESNUM(COLUMN,ROW),2)) THEN
               WRITE(30,5630) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         BPRAUNIV(COLUMN,ROW)
 5630          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,' IMP:N=1 U=',I3,
     c         '   $ Assembly lower end-fitting')
               LN=LN+1
            ELSE
               WRITE(30,5640) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*BPLEFTOPSURF),
     c         GTORSURF, BPRAUNIV(COLUMN,ROW)
 5640          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
               LN=LN+1
               WRITE(30,5650) LN, FRLEFML(COLUMN,ROW),
     c         (-1*LEFMAT(DESNUM(COLUMN,ROW),1)), (-1*GTBOTSURF),
     c         (-1*GTORSURF), BPRAUNIV(COLUMN,ROW)
 5650          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly lower end-fitting')
               LN=LN+1
            ENDIF
*     Write the upper end-fitting cell specification for this BPR universe.
            IF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,5660) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPRAUNIV(COLUMN,ROW)
 5660          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
            ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c      SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c      (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c      SURFVALUESPEC(UEFTOPSURF)).AND.
     c      (SURFVALUESPEC(GTTOPSURF).LE.
     c      SURFVALUESPEC(UEFBOTTOMSURF))) THEN
               WRITE(30,5670) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c         (-1*UEFTOPSURF), BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5670          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
               LN=LN+1
               WRITE(30,5680) LN, FRUEFML(COLUMN,ROW),
     c         (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c         (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c         BPRAUNIV(COLUMN,ROW)
 5680          FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c         ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
```

```
            LN=LN+1
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c     SURFVALUESPEC(GTTOPSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF))) THEN
              WRITE(30,5690) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5690        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,5700) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTORSURF),
     c        BPRAUNIV(COLUMN,ROW)
 5700        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
              WRITE(30,5710) LN, BMODML,
     c        (-1*MODDENSITY), BPCLADTOPSURF,
     c        (-1*GTTOPSURF), (-1*GTIRSURF),
     c        BPRAUNIV(COLUMN,ROW)
 5710        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).LE.
     c     SURFVALUESPEC(GTTOPSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).EQ.
     c     SURFVALUESPEC(UEFTOPSURF))) THEN
              WRITE(30,5720) LN, FRUEFML(COLUMN,ROW),
     c        (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c        (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5720        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
           IF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c     SURFVALUESPEC(GTTOPSURF)) THEN
              WRITE(30,5730) LN, BMODML,
     c        (-1*MODDENSITY), BPCLADTOPSURF,
     c        (-1*UEFTOPSURF), (-1*GTIRSURF),
     c        BPRAUNIV(COLUMN,ROW)
 5730        FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c        ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
              LN=LN+1
           ENDIF
           ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c     SURFVALUESPEC(UEFTOPSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).GT.
     c     SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c     (SURFVALUESPEC(GTTOPSURF).LT.
```

```
     c       SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,5740) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5740           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5750) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5750           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5760) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), BPCLADTOPSURF,
     c          (-1*UEFTOPSURF), (-1*BPCLADORSURF),
     c          BPRAUNIV(COLUMN,ROW)
 5760           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).EQ.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).LT.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,5770) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5770           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
                WRITE(30,5780) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5780           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ELSEIF ((SURFVALUESPEC(BPCLADTOPSURF).GT.
     c        SURFVALUESPEC(UEFBOTTOMSURF)).AND.
     c        (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c        SURFVALUESPEC(UEFTOPSURF)).AND.
     c        (SURFVALUESPEC(GTTOPSURF).EQ.
     c        SURFVALUESPEC(BPCLADTOPSURF))) THEN
                WRITE(30,5790) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), UEFBOTTOMSURF,
     c          (-1*UEFTOPSURF), GTORSURF, BPRAUNIV(COLUMN,ROW)
 5790           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          ' IMP:N=1 U=',I3,'   $ Assembly upper end-fitting')
                LN=LN+1
```

```
                WRITE(30,5800) LN, FRUEFML(COLUMN,ROW),
     c          (-1*UEFMAT(DESNUM(COLUMN,ROW),1)), GTTOPSURF,
     c          (-1*UEFTOPSURF), (-1*GTORSURF),
     c          BPCLADORSURF, BPRAUNIV(COLUMN,ROW)
 5800           FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c          1X,I4,' IMP:N=1 U=',I3,
     c          '   $ Assembly upper end-fitting')
                LN=LN+1
              ENDIF
*     Write the GT material cell in this BPR universe.
*     Determine if the GT material specification has
*     previously been defined.  If it has been previously defined, determine
*     the material specification label.
            CLADMLUNIQUE=.TRUE.
            LEAVE=.FALSE.
            IF ((COLUMN.NE.1).AND.(ROW.NE.1)) THEN
               DO 5820 RO=1,(ROW-1)
                  DO 5810 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 5810             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5820          CONTINUE
               IF (LEAVE.EQ..FALSE.) THEN
                  DO 5840 RO=ROW,ROW
                     DO 5830 CO=1,(COLUMN-1)
                        IF (DESNUM(CO,RO).NE.0) THEN
                           IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                     GTMAT(DESNUM(CO,RO))) THEN
                              CLADMLUNIQUE=.FALSE.
                              LEAVE=.TRUE.
                              GTML(COLUMN,ROW)=GTML(CO,RO)
                              EXIT
                           ENDIF
                        ENDIF
 5830                CONTINUE
                     IF (LEAVE.EQ..TRUE.) THEN
                        EXIT
                     ENDIF
 5840             CONTINUE
               ENDIF
            ELSEIF ((COLUMN.EQ.1).AND.(ROW.NE.1)) THEN
               DO 5860 RO=1,(ROW-1)
                  DO 5850 CO=1,50
                     IF (DESNUM(CO,RO).NE.0) THEN
```

```
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 5850             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5860          CONTINUE
            ELSEIF ((ROW.EQ.1).AND.(COLUMN.NE.1)) THEN
               DO 5880 RO=1,1
                  DO 5870 CO=1,(COLUMN-1)
                     IF (DESNUM(CO,RO).NE.0) THEN
                        IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.
     c                  GTMAT(DESNUM(CO,RO))) THEN
                           CLADMLUNIQUE=.FALSE.
                           LEAVE=.TRUE.
                           GTML(COLUMN,ROW)=GTML(CO,RO)
                           EXIT
                        ENDIF
                     ENDIF
 5870             CONTINUE
                  IF (LEAVE.EQ..TRUE.) THEN
                     EXIT
                  ENDIF
 5880          CONTINUE
            ENDIF
            IF (CLADMLUNIQUE.EQ..TRUE.) THEN
               GTML(COLUMN,ROW)=MN
*  Check Guide Tube Material
               IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                  DO 5890 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9300) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9301)
                        WRITE(200,7000)
                        WRITE(200,7001)
                        WRITE(200,7002)
                        WRITE(200,9302)
                        WRITE(200,7003)
                        WRITE(200,7004)
                        WRITE(200,7005)
                        WRITE(200,9303)
                        WRITE(200,9304)
                     ENDIF
 5890             CONTINUE
                ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     c          .EQ.2) THEN
                  DO 5900 C=1,2
```

```
                  IF (C.EQ.1) THEN
                     WRITE(200,9305) GTML(COLUMN,ROW)
                  ELSEIF (C.EQ.2) THEN
                     WRITE(200,9306)
                     WRITE(200,9307)
                     WRITE(200,9308)
                     WRITE(200,9309)
                     WRITE(200,9310)
                     WRITE(200,7006)
                     WRITE(200,7007)
                     WRITE(200,7008)
                     WRITE(200,9311)
                     WRITE(200,9312)
                     WRITE(200,7009)
                     WRITE(200,7010)
                     WRITE(200,7011)
                     WRITE(200,9313)
                     WRITE(200,7012)
                     WRITE(200,7013)
                     WRITE(200,7014)
                     WRITE(200,7015)
                  ENDIF
5900              CONTINUE
                ELSEIF (GTMAT(DESNUM(COLUMN,ROW))
     C          .EQ.3) THEN
                  DO 5910 C=1,2
                     IF (C.EQ.1) THEN
                        WRITE(200,9314) GTML(COLUMN,ROW)
                     ELSEIF (C.EQ.2) THEN
                        WRITE(200,9315)
                        WRITE(200,9316)
                        WRITE(200,9317)
                        WRITE(200,9318)
                        WRITE(200,7016)
                        WRITE(200,7017)
                        WRITE(200,7018)
                        WRITE(200,9319)
                        WRITE(200,9320)
                        WRITE(200,7019)
                        WRITE(200,7020)
                        WRITE(200,7021)
                        WRITE(200,9321)
                        WRITE(200,7022)
                        WRITE(200,7023)
                        WRITE(200,7024)
                        WRITE(200,7025)
                        WRITE(200,9322)
                        WRITE(200,9323)
                        WRITE(200,9324)
                        WRITE(200,9325)
                        WRITE(200,9326)
                        WRITE(200,9327)
                        WRITE(200,7026)
                        WRITE(200,9328)
```

```
                              WRITE(200,9329)
                              WRITE(200,9330)
                            ENDIF
 5910                     CONTINUE
                        ENDIF
                     MN=MN+1
                  ENDIF
                  IF (GTMAT(DESNUM(COLUMN,ROW)).EQ.1) THEN
                     CLADRHO=6.56
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.2) THEN
                     CLADRHO=7.90
                  ELSEIF (GTMAT(DESNUM(COLUMN,ROW)).EQ.3) THEN
                     CLADRHO=8.19
                  ENDIF
                  WRITE(30,5920) LN, GTML(COLUMN,ROW), (-1*CLADRHO),
     c            GTIRSURF,
     c            (-1*GTORSURF), (-1*GTTOPSURF), GTBOTSURF,
     c            BPRAUNIV(COLUMN,ROW)
 5920             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Guide tube region')
                  LN=LN+1
*     Write the moderator cells within the GT in this BPR universe.
                  IF (SURFVALUESPEC(BPCLADTOPSURF).GE.
     c            SURFVALUESPEC(GTTOPSURF)) THEN
                     WRITE(30,5930) LN, BMODML, (-1*MODDENSITY),
     c               (-1*GTIRSURF),
     c               BPOCORSURF, (-1*GTTOPSURF), GTBOTSURF,
     c               BPRAUNIV(COLUMN,ROW)
 5930                FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c               ' IMP:N=1 U=',I3,
     c               '   $ Borated moderator inside guide tube')
                     LN=LN+1
                  ELSEIF (SURFVALUESPEC(BPCLADTOPSURF).LT.
     c            SURFVALUESPEC(GTTOPSURF)) THEN
                     WRITE(30,5940) LN, BMODML, (-1*MODDENSITY),
     c               (-1*GTIRSURF),
     c               BPOCORSURF, (-1*BPCLADTOPSURF), GTBOTSURF,
     c               BPRAUNIV(COLUMN,ROW)
 5940                FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,1X,I4,
     c               ' IMP:N=1 U=',I3,
     c               '   $ Borated moderator inside guide tube')
                     LN=LN+1
                  ENDIF
                  WRITE(30,5950) LN, BMODML, (-1*MODDENSITY),
     c            (-1*BPOCORSURF), (-1*BPCLADBOTTOMSURF), GTBOTSURF,
     c            BPRAUNIV(COLUMN,ROW)
 5950             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Borated moderator inside guide tube')
                  LN=LN+1
*     Loop through the regions above the BPR (i.e. the appropriate upper core
regions)
*     Define the upper region lower surface.
                     DO 5990 REGION=1,NUMREGABOVEBPRA
*     Determine the current upper region's lower surface specification.
```

```
                  IF (REGION.EQ.1) THEN
                     REGIONTOPSURF=SYSTEMTOP
                     CURRENTSURF=SURFVALUESPEC(SYSTEMTOP)-
     c               REGABOVEBPRA(REGION,1)
                  ENDIF
                  CURRENTSURF=SURFVALUESPEC(REGIONTOPSURF)-
     c            REGABOVEBPRA(REGION,1)
                  IF (REGION.EQ.NUMREGABOVEBPRA) THEN
                     REGIONBOTTOMSURF=UEFTOPSURF
                  ELSE
                  CURRENTSURFLABEL=0
                  DO 5960 V=1,(SN-1)
                     IF (SURFTYPESPEC(V).EQ.'PZ') THEN
                IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                           CURRENTSURFLABEL=V
                           EXIT
                        ENDIF
                     ENDIF
 5960             CONTINUE
                  IF (CURRENTSURFLABEL.EQ.0) THEN
                     REGIONBOTTOMSURF=SN
                     SURFTYPESPEC(SN)='PZ'
                     SURFVALUESPEC(SN)=CURRENTSURF
                     SN=SN+1
                  ELSE
                     REGIONBOTTOMSURF=CURRENTSURFLABEL
                  ENDIF
                  ENDIF
*     Write the cell specification for the BPR universe upper region.
               IF (REGION.EQ.1) THEN
                  WRITE(30,5970) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5970             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ELSE
                  WRITE(30,5980) LN, FRUREGIONML(COLUMN,ROW,REGION),
     c            (-1*REGABOVEBPRA(REGION,2)), (-1*REGIONTOPSURF),
     c            REGIONBOTTOMSURF, BPRAUNIV(COLUMN,ROW), REGION
 5980             FORMAT(T1,I4,T6,I4,T11,F8.5,T25,I4,1X,I4,
     c            ' IMP:N=1 U=',I3,'   $ Upper core region ',I2)
                  LN=LN+1
                  REGIONTOPSURF=REGIONBOTTOMSURF
               ENDIF
 5990          CONTINUE
            SPACHEIGHT=0.0
*     Loop through the spacer and moderator regions along the axial
*     length of the BPR (from top to bottom).
            DO 6000 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
               SPACHEIGHT=SPACHEIGHT+SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
 6000       CONTINUE
            DO 6080 SPN=1,NUMOFSPACERS(DESNUM(COLUMN,ROW))
```

```
*     Define the homogenized spacer region bounding surfaces.
            IF (SPN.EQ.1) THEN
               SPACERTOPSURF=UEFBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(UEFBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 6010 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 6010          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 6020 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 6020          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF ((SPN.NE.1).AND.(SPN.NE.
     c      NUMOFSPACERS(DESNUM(COLUMN,ROW)))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 6030 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
```

```
 6030          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               CURRENTSURF=SPACERDIST(DESNUM(COLUMN,ROW),(SPN+1))
               CURRENTSURFLABEL=0
               DO 6040 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
            IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 6040          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  WATERREGIONBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  WATERREGIONBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
            ELSEIF (SPN.EQ.NUMOFSPACERS(DESNUM(COLUMN,ROW))) THEN
               SPACERTOPSURF=WATERREGIONBOTTOMSURF
               CURRENTSURF=SURFVALUESPEC(WATERREGIONBOTTOMSURF)-
     c         SPACERHEIGHT(DESNUM(COLUMN,ROW),SPN)
               CURRENTSURFLABEL=0
               DO 6050 V=1,(SN-1)
                  IF (SURFTYPESPEC(V).EQ.'PZ') THEN
             IF (ABS(SURFVALUESPEC(V)-CURRENTSURF).LT.(0.0001)) THEN
                        CURRENTSURFLABEL=V
                        EXIT
                     ENDIF
                  ENDIF
 6050          CONTINUE
               IF (CURRENTSURFLABEL.EQ.0) THEN
                  SPACERBOTTOMSURF=SN
                  SURFTYPESPEC(SN)='PZ'
                  SURFVALUESPEC(SN)=CURRENTSURF
                  SN=SN+1
               ELSE
                  SPACERBOTTOMSURF=CURRENTSURFLABEL
               ENDIF
               WATERREGIONTOPSURF=SPACERBOTTOMSURF
               WATERREGIONBOTTOMSURF=NODEBOTTOMSURF
            ENDIF
*     Write the current homogenized spacer region cell in this BPR universe.
            WRITE(30,6060) LN, HOMOSPACMLNUM(DESNUM(COLUMN,ROW),SPN),
```

```
     c       (-1*HOMOSPACERDEN(DESNUM(COLUMN,ROW),SPN)), GTORSURF,
     c       (-1*SPACERTOPSURF), SPACERBOTTOMSURF, BPRAUNIV(COLUMN,ROW),
     c       SPN
 6060        FORMAT(T1,I4,T6,I4,T11,G14.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Homogenized region for spacer ',I2)
             LN=LN+1
*     Write the water region cell below the current homogenized spacer cell
in this BPR universe.
             WRITE(30,6070) LN, BMODML, (-1*MODDENSITY), GTORSURF,
     c       (-1*WATERREGIONTOPSURF), WATERREGIONBOTTOMSURF,
     c       BPRAUNIV(COLUMN,ROW)
 6070        FORMAT(T1,I4,T6,I4,T11,F10.8,T25,I4,1X,I4,1X,I4,
     c       ' IMP:N=1 U=',I3,'   $ Borated moderator')
             LN=LN+1
 6080      CONTINUE
         ENDIF
       ENDIF
 9300                  FORMAT(T1,'M',I4,T9,' 8016.50c     -0.120',
     c                       '          $ Zirc-4 Cladding')
 9301                       FORMAT(T9,'24050.60c     -0.004')
 7000                       FORMAT(T9,'24052.60c     -0.084')
 7001                       FORMAT(T9,'24053.60c     -0.010')
 7002                       FORMAT(T9,'24054.60c     -0.002')
 9302                       FORMAT(T9,'26054.60c     -0.011')
 7003                       FORMAT(T9,'26056.60c     -0.184')
 7004                       FORMAT(T9,'26057.60c     -0.004')
 7005                       FORMAT(T9,'26058.60c     -0.001')
 9303                       FORMAT(T9,'40000.60c     -98.180')
 9304                       FORMAT(T9,'50000.35c     -1.400')
 9305                  FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
     c                       '          $ SS304 Cladding')
 9306                       FORMAT(T9,'7014.50c      -0.100')
 9307                       FORMAT(T9,'14000.50c     -0.750')
 9308                       FORMAT(T9,'15031.50c     -0.045')
 9309                       FORMAT(T9,'16032.50c     -0.030')
 9310                       FORMAT(T9,'24050.60c     -0.793')
 7006                       FORMAT(T9,'24052.60c     -15.903')
 7007                       FORMAT(T9,'24053.60c     -1.838')
 7008                       FORMAT(T9,'24054.60c     -0.466')
 9311                       FORMAT(T9,'25055.50c     -2.000')
 9312                       FORMAT(T9,'26054.60c     -3.918')
 7009                       FORMAT(T9,'26056.60c     -63.156')
 7010                       FORMAT(T9,'26057.60c     -1.472')
 7011                       FORMAT(T9,'26058.60c     -0.200')
 9313                       FORMAT(T9,'28058.60c     -6.234')
 7012                       FORMAT(T9,'28060.60c     -2.465')
 7013                       FORMAT(T9,'28061.60c     -0.109')
 7014                       FORMAT(T9,'28062.60c     -0.350')
 7015                       FORMAT(T9,'28064.60c     -0.092')
 9314                  FORMAT(T1,'M',I4,T9,'6000.50c      -0.080',
     c                       '          $ Inconel Cladding')
 9315                       FORMAT(T9,'14000.50c     -0.350')
 9316                       FORMAT(T9,'15031.50c     -0.015')
 9317                       FORMAT(T9,'16032.50c     -0.015')
```

```
9318                    FORMAT(T9,'24050.60c     -0.793')
7016                    FORMAT(T9,'24052.60c     -15.903')
7017                    FORMAT(T9,'24053.60c     -1.838')
7018                    FORMAT(T9,'24054.60c     -0.466')
9319                    FORMAT(T9,'25055.50c     -0.350')
9320                    FORMAT(T9,'26054.60c     -0.958')
7019                    FORMAT(T9,'26056.60c     -15.442')
7020                    FORMAT(T9,'26057.60c     -0.360')
7021                    FORMAT(T9,'26058.60c     -0.049')
9321                    FORMAT(T9,'28058.60c     -35.382')
7022                    FORMAT(T9,'28060.60c     -13.993')
7023                    FORMAT(T9,'28061.60c     -0.616')
7024                    FORMAT(T9,'28062.60c     -1.989')
7025                    FORMAT(T9,'28064.60c     -0.520')
9322                    FORMAT(T9,'5010.50c      -1.078E-3')
9323                    FORMAT(T9,'5011.56c      -4.925E-3')
9324                    FORMAT(T9,'13027.50c     -0.500')
9325                    FORMAT(T9,'22000.50c     -0.900')
9326                    FORMAT(T9,'27059.50c     -1.000')
9327                    FORMAT(T9,'29063.60c     -0.205')
7026                    FORMAT(T9,'29065.60c     -0.095')
9328                    FORMAT(T9,'41093.50c     -2.563')
9329                    FORMAT(T9,'42000.50c     -3.050')
9330                    FORMAT(T9,'73181.50c     -2.563')

     RETURN
     END
```

This attachment contains the MACE input decks for the reactivity calculations for McGuire Unit 1. The input decks are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.14). The information contained in this hard-copy representation of Attachment II is a listing of the various MACE input deck files and their attributes. The file sizes listed in the following table are the file sizes as contained on the attachment tape (Ref. 7.14). The tape containing Attachment II was written using the HP Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size on Tape (bytes)	Date File Copied to Tape
mg1i1a.txt	ASCII	77,761	03/09/98
mg1i2a.txt	ASCII	80,112	03/09/98
mg1i2b.txt	ASCII	80,104	03/09/98
mg1i2c.txt	ASCII	80,104	03/09/98
mg1i2d.txt	ASCII	80,104	03/09/98
mg1i3a.txt	ASCII	80,114	03/09/98
mg1i3b.txt	ASCII	80,106	03/09/98
mg1i3c.txt	ASCII	80,106	03/09/98
mg1i3d.txt	ASCII	80,106	03/09/98
mg1i4a.txt	ASCII	74,993	03/09/98
mg1i4b.txt	ASCII	74,985	03/09/98
mg1i4c.txt	ASCII	74,985	03/09/98
mg1i4d.txt	ASCII	74,985	03/09/98
mg1i5a.txt	ASCII	74,992	03/09/98
mg1i5b.txt	ASCII	74,984	03/09/98
mg1i5c.txt	ASCII	74,984	03/09/98
mg1i5d.txt	ASCII	74,984	03/09/98
mg1i6a.txt	ASCII	74,992	03/09/98
mg1i6b.txt	ASCII	74,984	03/09/98
mg1i6c.txt	ASCII	74,984	03/09/98
mg1i6d.txt	ASCII	74,984	03/09/98

This attachment contains the MCNP input decks for the reactivity calculations for McGuire Unit 1. The input decks are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.14). The information contained in this hard-copy representation of Attachment III is a listing of the various MCNP input deck files and their attributes. The file sizes listed in the following table are the file sizes as contained on the attachment tape (Ref. 7.14). The tape containing Attachment III was written using the HP Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size on Tape (bytes)	Date File Copied to Tape
mg1i1a	ASCII	447,706	03/09/98
mg1i2a	ASCII	1,311,190	03/09/98
mg1i2b	ASCII	869,333	03/09/98
mg1i2c	ASCII	707,093	03/09/98
mg1i2d	ASCII	664,213	03/09/98
mg1i3a	ASCII	1,664,469	03/09/98
mg1i3b	ASCII	1,016,911	03/09/98
mg1i3c	ASCII	765,646	03/09/98
mg1i3d	ASCII	700,116	03/09/98
mg1i4a	ASCII	1,411,210	03/09/98
mg1i4b	ASCII	951,097	03/09/98
mg1i4c	ASCII	780,745	03/09/98
mg1i4d	ASCII	735,721	03/09/98
mg1i5a	ASCII	1,824,785	03/09/98
mg1i5b	ASCII	1,099,817	03/09/98
mg1i5c	ASCII	840,233	03/09/98
mg1i5d	ASCII	771,625	03/09/98
mg1i6a	ASCII	1,873,399	03/09/98
mg1i6b	ASCII	1,099,767	03/09/98
mg1i6c	ASCII	840,183	03/09/98
mg1i6d	ASCII	771,575	03/09/98

This attachment contains the MCNP output files for the reactivity calculations for McGuire Unit 1. The output files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.14). The information contained in this hard-copy representation of Attachment IV is a listing of the various MCNP output files and their attributes. The file sizes listed in the following table are the file sizes as contained on the attachment tape (Ref. 7.14). The tape containing Attachment IV was written using the HP Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size on Tape (bytes)	Date File Copied to Tape
mg1i1a.O	ASCII	43,111,556	03/09/98
mg1i2a.O	ASCII	66,701,992	03/09/98
mg1i2b.O	ASCII	63,889,343	03/09/98
mg1i2c.O	ASCII	62,841,031	03/09/98
mg1i2d.O	ASCII	62,567,963	03/09/98
mg1i3a.O	ASCII	68,916,021	03/09/98
mg1i3b.O	ASCII	64,792,748	03/09/98
mg1i3c.O	ASCII	63,176,049	03/09/98
mg1i3d.O	ASCII	62,756,514	03/09/98
mg1i4a.O	ASCII	203,245,806	03/09/98
mg1i4b.O	ASCII	200,317,991	03/09/98
mg1i4c.O	ASCII	199,219,539	03/09/98
mg1i4d.O	ASCII	198,931,945	03/09/98
mg1i5a.O	ASCII	205,844,764	03/10/98
mg1i5b.O	ASCII	201,226,103	03/10/98
mg1i5c.O	ASCII	199,558,838	03/10/98
mg1i5d.O	ASCII	199,121,598	03/10/98
mg1i6a.O	ASCII	206,151,382	03/10/98
mg1i6b.O	ASCII	201,227,184	03/10/98
mg1i6c.O	ASCII	199,558,111	03/10/98
mg1i6d.O	ASCII	199,119,635	03/10/98

This attachment contains the MCNP input and output files for the k_{inf} calculations performed to evaluate the spacer grid modeling approximation effects. The input and output files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.14). The information contained in this hard-copy representation of Attachment V is a listing of the various MCNP input and output files and their attributes. The file sizes listed in the following table are the file sizes as contained on the attachment tape (Ref. 7.14). The tape containing Attachment V was written using the HP Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size on Tape (bytes)	Date File Copied to Tape
kinf1	ASCII	63,830	03/10/98
kinf1.O	ASCII	1,552,684	03/10/98
kinf2	ASCII	62,916	03/10/98
kinf2.O	ASCII	1,556,610	03/10/98

This attachment contains the MCNP input and output files for the corrected Principal Isotope reactivity calculations. The input and output files are contained on an attachment tape of this calculation file (the attachment tape has been moved to Reference 7.14). The information contained in this hard-copy representation of Attachment VI is a listing of the various MCNP input and output files and their attributes. The file sizes listed in the following table are the file sizes as contained on the attachment tape (Ref. 7.14). The tape containing Attachment VI was written using the HP Colorado Model T1000e External Parallel Port Backup System for personal computers.

Filename	File Type	File Size on Tape (bytes)	Date File Copied to Tape
mg1i2b	ASCII	869,338	06/03/98
mg1i2b.O	ASCII	8,373,407	06/03/98
mg1i3b	ASCII	1,016,916	06/03/98
mg1i3b.O	ASCII	9,278,508	06/03/98
mg1i4b	ASCII	951,102	06/03/98
mg1i4b.O	ASCII	8,502,177	06/03/98
mg1i5b	ASCII	1,099,822	06/03/98
mg1i5b.O	ASCII	9,413,301	06/03/98
mg1i6b	ASCII	1,099,772	06/03/98
mg1i6b.O	ASCII	9,412,603	06/03/98