サンプルユーザーコード

平山　英夫、波戸　芳仁
KEK, 高エネルギー加速器研究機構

Sample user codes
(テキストが用意されているもの)

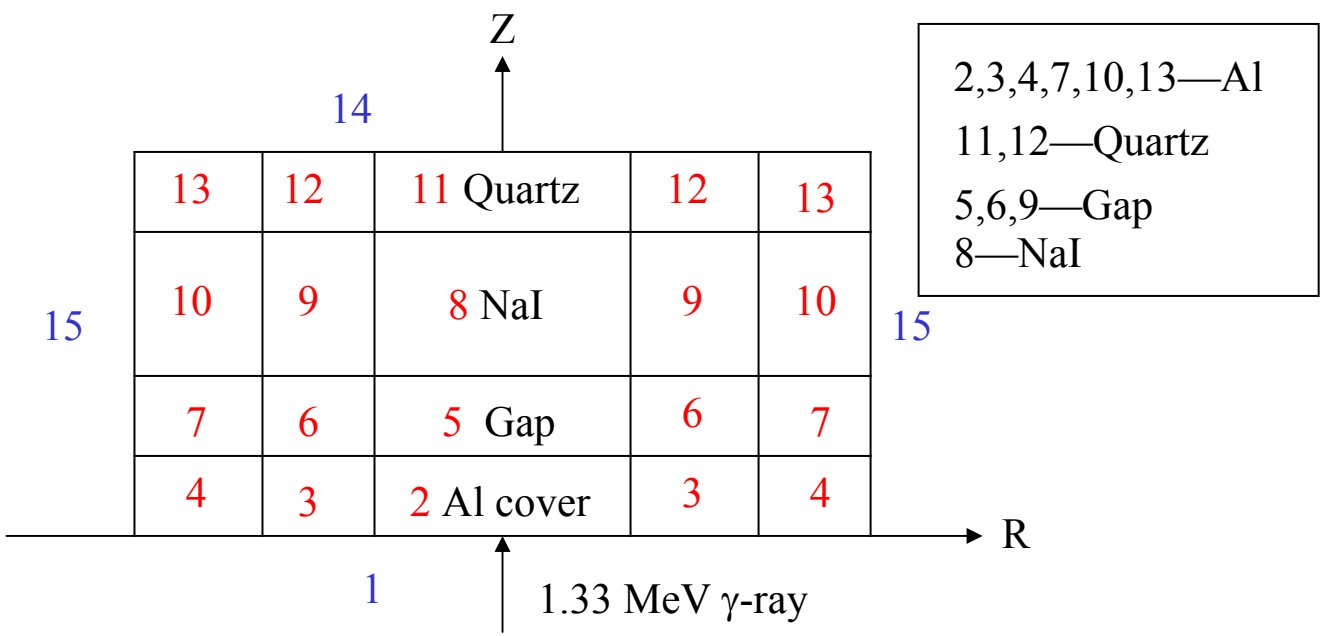
ユーザーコード名	形状	計算課題	対象飛跡表示システム
ucrz_nai.f	円筒平板	NaI検出器のレスポンス	使用せず
uccg_nai.f	CG	同　　上	使用せず
ucxyz_phantom.f	ボクセル	ファントム中の線量分布	PICT32
uccg_phantom.f	CG	同　　上	CGview

その他のサンプルユーザーコード

- メインプログラムで、物質名、リージョンへの物質の割り当て、ジオメトリー、入射粒子の条件を指定する形式
  - ucshield.f -- ucshield.mor に対応
  - ucphantom_rec.f – ucphantom_rec.morに対応
- 円筒平板形状
  - ucrz_sampl4.f – ucsampl4.morを円筒平板形状に
- ボクセル形状
  - ucxyz_dose.f – xyzdose.mor に対応

ucrz_nai.f

- 円筒ー平板形状でのNaI検出器のレスポンス計算



include文

- egs5で使用されているcommonや、そこで使用されている変数の配列の大きさは、別のファイルにparameter文で決め、include文により取り込まれる。
  - egs5に直接関係するinclude関係のファイルは、egsディレクトリーのincludeディレクトリ
  - ユーザー固有のユーザーコードに関連したinclude関係ファイルは、ユーザーのワーキングディレクトリーのサブディレクトリーuser_auxcommonディレクトリー

```
include 'include/egs5_h.f'              ! Main EGS "header" file

 include 'include/egs5_edge.f'
 include 'include/egs5_media.f'
 include 'include/egs5_misc.f'
 include 'include/egs5_switches.f'
 include 'include/egs5_uphiot.f'
 include 'include/egs5_useful.f'
 include 'include/randomm.f'
```

egs5 common に含まれる変数をメインプログラム等のプログラム単位で使用する場合は、include文で当該commonを指定

```
include 'user_auxcommons/aux_h.f'   ! Auxiliary-code "header" file

 include 'user_auxcommons/cyldta.f'
 include 'user_auxcommons/edata.f'
 include 'user_auxcommons/etaly1.f'
 include 'user_auxcommons/georz.f'
 include 'user_auxcommons/instuf.f'
 include 'user_auxcommons/lines.f'
 include 'user_auxcommons/pladta.f'
 include 'user_auxcommons/watch.f'
```

ジオメトリー関係等、直接egs5と関係しないcommon

In include/egs5_h.f

```
! Maximum number of regions allocated
      integer MXREG
parameter (MXREG = 10649)
```

リージョン数を増やしたい場合には、この数値を変更する。

include/egs5_misc.f

```
 common/MISC/                          ! Miscellaneous COMMON
     * rhor(MXREG), dunit,
     * med(MXREG),iraylr(MXREG),lpolar(MXREG),incohr(MXREG),
     * iprofr(MXREG),impacr(MXREG),
     * kmpi,kmpo,noscat

      real*8
     * rhor,dunit

      integer
     * med,iraylr,lpolar,incohr,iprofr,impacr,kmpi,kmpo,noscat
```

```
 common/totals/                          ! Variables to score
* depe,deltae,spg(1,50),spe(1,50),spp(1,50),nreg
 real*8 depe,deltae,spg,spe,spp
 integer nreg

   real*8                                 ! Local variables
* availke,avpe,avph,avspe,avspg,avspp,avte,ekin,etot,
* desci2,pef,rnnow,sigpe,sigph,sigspe,sigspg,sigspp,
* sigte,tef,totke,wtin,wtsum

 real*8
* ph(50),phpb(50,50),spgpb(1,50,50),spepb(1,50,50),
* spppb(1,50,50),pefpb(50),tefpb(50)

 real                                   ! Local variables
* elow,eup,rdet,rtcov,rtgap,tcov,tdet,tgap

 real
* tarray(2),tt,tt0,tt1,cputime

 integer
* i,icases,idin,ie,imed,ireg,isam,isot,
* j,k,n,nbatch,ncaspb,nd,ndet,nlist,nofbat
```

Open文

- ユーザーコードから、pegsを実行するのに伴い、ユニット7-26は、pegsで close されることから、メインプログラムで open していても、pegs実行後に、再度 open することが必要となる。そのため、ユニット7-26の使用を避ける方が良い。
- 飛跡情報を出力するplotxyz.fのユニットは、9から39に変更

Subroutine getrz の call

- サブルーティンの名称、機能をどの様に設定するかは、個々のユーザーコードにより異なるが、最低必要な機能は、pegsの実行と、subroutine hatch を call することである。
- getrzでは、使用する物質の指定、カットオフエネルギー、各種オプション、入射粒子等の設定を全てこのサブルーティンにおいて、ユニット4から読み込むデータで設定する様にしている。
- uin=vin=win=0.0の時は、メインプログラムで、等方線源であることを示すフラグisotを1にする。

Call shower

- 各バッチ毎に、ncaspb ヒストリーだけ subroutine shower を call する。この操作を、設定したバッチ回数(nbatch)繰り返す。
- 線源のエネルギーは、ユニット4から読み込んだデータに基づき決定する。
- 各ヒストリー毎に、NaI領域での吸収エネルギーの有無を調べ、吸収エネルギーがある場合には、全検出効率の数に、そのエネルギーが、入射粒子の99.9%以上の時は、ピーク検出効率の数に加える。また、吸収エネルギーの値により、波高分布のどの位置に属するかを調べる。

線源粒子のエネルギー決定

- isamp=0
  - 単一エネルギー：getrzで読み込んだekin
- isamp=1
  - 離散エネルギー(RIからのγ線）
- isamp=2
  - Getrzで読み込んだpdfから作成した、cdfを用いて、サンプリング
- isamp=3
  - 対象となるエネルギー領域で、一様サンプリングを行い、対応するエネルギーのpdfをウエイトとして適用

統計的な誤差評価

- x をモンテカルロ計算によって求める量とする誤差を評価するのに便利な2つの方法がある
- MCNPで使用している方法
  - 計算は N 個の“入射” 粒子について行われ、x_i は、i-番目のヒストリーの結果であるとする

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$ x_i の平均値

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - x)^2 \approx \overline{x^2} - (\bar{x})^2 ; (\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2)$$ Variance associated with distribution of x_i

$$s_{\bar{x}}^2 = \frac{1}{N}s^2 \approx \frac{1}{N}[\overline{x^2} - \bar{x}^2]$$ Variance associated with distribution of $\bar{x}$

$$R = \frac{s_{\bar{x}}}{\bar{x}} \approx [\frac{1}{N}(\frac{\overline{x^2}}{\bar{x}^2} - 1)]^{1/2}$$ Statistical error

MORSE-CGで使用している方法

- 計算は N 個の“入射” 粒子について行われ、x_i は、i-番目のヒストリーの結果であるとする
- “N” ヒストリーを、それぞれ N/n ヒストリーのn 個のバッチに分割する
- 各バッチ毎に得られた値を x_jとする

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$$ Mean value of x_j

$$s_x^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - x)^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j^2 - \bar{x}^2)$$ Variance associated with distribution of x_j

$$s_{\bar{x}}^2 = \frac{s_x^2}{n}$$ Standard variance of the mean

$$FSD = \frac{s_{\bar{x}}}{\bar{x}}$$ Fractional standard deviation

ucrz_nai.f で使用している誤差評価の方法

- MORSE-CG で使用している方法により FSD を求める
- 平均値 $\bar{x}$ は、各バッチの終わりで計算する

```
!  Calculate average value for this BATCH
     do ie=1,50
       phpb(ie,nofbat) = ph(ie) /ncaspb
       ph(ie)=0.D0
     end do
     pefpb(nofbat)=pef / ncaspb
     tefpb(nofbat)=tef /ncaspb
     pef=0.D0
     tef=0.D0
     do nd=1,ndet
       do ie=1,50
         spgpb(nd,ie,nofbat)=spg(nd,ie)/ncaspb !photon spectrum
         spepb(nd,ie,nofbat)=spe(nd,ie)/ncaspb !electron spectrum
         spppb(nd,ie,nofbat)=spp(nd,ie)/ncaspb !positron spectrum
         spg(nd,ie)=0.D0
         spe(nd,ie)=0.D0
         spp(nd,ie)=0.D0
       end do
     end do
```

波高分布

ピーク検出効率

全検出効率

NaIに入射した粒子のスペクトル

結果の解析

- バッチ毎の平均値から、平均値とFSDを求め、出力する

```
!     ---------------
!     Peak efficiency
!     ---------------
      avpe = 0.D0
      desci2 = 0.D0
      do j = 1, nbatch
        avpe = avpe + pefpb(j)/nbatch
        desci2 = desci2 + pefpb(j)*pefpb(j)/nbatch
      end do
      sigpe = sqrt((desci2 - avpe*avpe)/(nbatch-1))
      avpe = avpe*100.0
      sigpe = sigpe*100.0
      write(6,210) avpe,sigpe
210   FORMAT(' Peak efficiency =',G15.5,'+-',G15.5,' %')
```

Subroutine getrz

・円筒平板の形状を含め、計算に必要な情報をデータファイルから入力

・pegsをcallして、物質データを計算し、その後 Subroutine hatch を call

```
! -----------------
! Record 1: title
! -----------------
      read(4,101) title
 101  FORMAT(80A1)
      write(6,102) title
 102  FORMAT(' TITLE:'/1X,80A1/)

! ------------------
! Record 2: nmed
! ------------------
      read(4,*) nmed
      if (nmed .gt. MXMED) then
        write(6,104) nmed
 104    FORMAT(' *** Stopped in GetRZ with nmed=',I5,' > MXMED')
        stop
      end if
      write(6,105) nmed
 105  FORMAT(' nmed=',I5,/)
```

タイトル（80文字）

使用する物質数

Parameter 文で設定した値を超えていないことの確認

```
! -------------------
! Record 3: media
! -------------------
      do i=1,nmed
        read(4,106) (media(j,i),j=1,24)
 106    FORMAT(24A1)
        write(6,107) i,(media(j,i),j=1,24)
 107    FORMAT(' MEDIUM=',I5,' ==> ',24A1)
      end do

! --------------------
! Record 4: ncyl, nplan
! --------------------
      read(4,*) ncyl, nplan

      if (ncyl .gt. MXCYLS) then
        write(6,114) ncyl
 114    FORMAT(' *** Stopped in getrz with ncyl=',I5,' > MXCYLS')
        stop
      end if
```

使用する物質名を、1行に1つずつ24文字以内で指定。

pegsで作成する物質に含まれていなければならない。

円筒の数(ncyl)

平板の数(nplan)

円筒の数が、parameter 文で設定した値を超えていないことの確認

平板の数についても同様に確認

```
!     -------------------------
      nreg = (nplan-1)*ncyl+3
      irz = nreg - 3
!     -------------------------

      if (nreg .gt. MXREG) then
        write(6,118) nreg
 118    FORMAT(' *** Stopped in getrz with nreg=',I5,' > MXREG')
        stop
      end if
      write(6,119) nreg
 119  FORMAT(/,' number of region (nreg) =',I5,/,
     *        ' nreg includs front, back and outside cylinder')
```

円筒と平板の数から、必要なリージョン数(nreg)を計算。最初の平板の手前、最後の平板の外側、最後の円筒の外側を別なリージョンとしている。

リージョン数が、parameter 文で設定した値を超えていないことを確認

```
! --------------
! Record 5: cyrad
! --------------
      write(6,120)
 120  FORMAT(/,' Input radius of cylinder:',/)

      do i=1,ncyl
        read(4,*) cyrad(i)
        cyrad2(i) = cyrad(i)**2
        write(6,122) i,cyrad(i)
 122      FORMAT(5X,'i=',I3,5X,'cyrad=',G15.7,' cm')
      end do

! -------------
! Record 6: tpl
! -------------
      write(6,127)
 127  FORMAT(/,' Input boundaries in the Z direction:',/)

      do k=1,nplan
        read(4,*) zpl(k)
        write(6,129) k,zpl(k)
 129    FORMAT(5X,'k=',I3,5X,'zpl=',G15.7,' cm')
      end do
```

円筒の半径を、小さいものから順に入力

howfarで使用する半径の二乗を計算

Z-軸に垂直な平板とZ-軸との交点の座標を小さい側から順に

```
 read(4,142) medtmp,rhotmp,ecutin,pcutin
142      FORMAT(I10,3F10.3)
         if (medtmp.ne.0) then
! ---------------------------------------
! Record 7a: ipeangsw,iedgesw,iraysw,ipolarsw,
!            incohrsw,iprofrsw,impacrsw
! ---------------------------------------
           read(4,145) ipeangsw,iedgesw,iraysw,ipolarsw,incohrsw,
     *     iprofrsw,impacrsw
145        FORMAT(7I5)
```

最初に、各Z-ビン毎に、同じ物質を割り当てる。物質番号、密度、カットオフエネルギーを入力

密度、カットオフエネルギーは、0.0 の時は、デフォルト値を使用

物質が真空 (0) でない場合には、各種オプションを設定

リージョン毎に設定できるオプション

iphter(irl)	Switches for PE-angle sampling
iedgfl(irl)	K & L-edge fluorescence
iraylr(irl)	Rayleigh scattering
lpolar(irl)	Linearly-polarized photon scattering
incohr(irl)	S/Z rejection
iprofr(irl)	Doppler broadening
impacr(irl)	Electron impact ionization

```
! ------------------------------------------------------------------
!  Record 8  nzbin, nrbin, meptmp, rhotmp, ecutin, pcutin
!------------------------------------------------------------------
!     ------------------------------------------
!     Check exception. nzbin=0 means end.
!     ------------------------------------------
160    continue
      read(4,162) nzbin,nrbin,medtmp,rhotmp,ecutin,pcutin
162    FORMAT(3I5,3F10.3)
      if(nzbin .eq. 0) go to 170
!     ----------------
!     Set exception.
!     ----------------
      irl=(nzbin-1)*ncyl+nrbin+1
      med(irl)=medtmp
```

Z-ビン毎に設定した物質と、条件が異なるリージョンの物質、密度、カットオフエネルギーの指定

Z-ビンと、R-ビンで指定

指定が終了した時は、Z-ビンの0を指定

Z-ビン及びR-ビンからリージョン番号を求め、物質番号を割り当てる。

真空でない場合は、各種オプションのフラグを入力し、設定する。

```
! --------------------
! Record 9: xin,yin,zin
! --------------------
      read(4,*) xin,yin,zin

! --------------
! Record 10: irin
! --------------
      read(4,*) irin

! ---------------------
! Record 11: uin,vin,win
! ---------------------
      read(4,*) uin,vin,win
```

線源粒子の入射位置

線源粒子の入射リージョン

入射粒子の方向余弦

uin=vin=win=0.0 の時は、

等方線源

RANECE用の2つのseedsの指定

前の計算の最後のseedsを設定すると、接続計算となる。
異常が生じたヒストリーの最初のseedsを、ixx と jxxに設定すれば、異常の原因を調べやすくなる。

```
! -----------------
! Record 12: ixx,jxx
! -----------------
      read(4,*) ixx,jxx
      if (ixx .eq. 0)  ixx = 123457            ! Default seed
      if (jxx .eq. 0)  jxx = 654321            ! Default seed
      write(6,210) ixx,jxx
210   FORMAT(/,' ixx=',I12,5X,'jxx=',I12,
     *       ' (starting random-number seeds)')

!     ---------------------------------
!     Save the starting random-number seeds
!     ---------------------------------
      iseed1=ixx
      iseed2=jxx
```

```
! ----------------
! Record 13: ncases
! ----------------
      read(4,*) ncases
      write(6,220) ncases
220   FORMAT(/,' ncases=',I12)

! --------------------------
! Record 14: ekein,iqin,isamp
! --------------------------
      read(4,*) ekein,iqin,isamp
```

ヒストリー数

線源の最大エネルギー

電荷

線源の種類

Isamp=0 : ekein の単一エネルギー

=1 : 離散エネルギーからサンプリング（RIからのγ線等）

=2 : cdf からサンプリング (pdfを入力し、cdfを計算）

=3 : エネルギーは、対象領域で一様にサンプリングし、

pdf をウエイトとして使用

```
! ---------------------
! Record 15: iwatch
! ---------------------
      read(4,*) iwatch

! -----------------------------------
! Record 16: ibrdst,iprdst,ibrspl,nbrspl
! -----------------------------------
      read(4,*) ibrdst,iprdst,ibrspl,nbrspl

      if (ibrspl .gt. 0) then
        if (nbrspl .gt. 0) then
          fbrspl = 1.0/float(nbrspl)
        else
          write(6,420) ibrspl,nbrspl
420       FORMAT(//,' Stopped in GetRZ with IBRSPL=',I5,' and NBRSPL=',
     *    I5)
          stop
        end if
      end if
```

Iwatch option の設定: 0=off, 1=each interaction, 2=each step

制動輻射角度分布オプション!
brdst=0 No (use default: theta=m/E)
1 Yes (recommended)

電子対生角度分布オプション
iprdst=0 No (use default: theta=m/E)
1 Yes (low-order distribution)
2 Yes (recommended)

スプリッティングの設定

ibrspl not 0 : nbrsplにスプリット

```
! ------------------------------
! Record 17: estepe,estepe2
! ------------------------------
      read(4,*) estepe, estepe2
```

荷電粒子輸送に使用するパラメータ

Alex 及び Scot の講義

ausgab の機能

- ausgab は、ユーザーが得たい情報を記録するサブルーチンである
- NaI検出器中での沈着エネルギーの記録

```
!    ------------------------------------------
!    Score energy deposition inside NaI detector
!    ------------------------------------------
     if (med(irl). eq. 1) then
       depe = depe + edepwt
```

当該リージョンの物質番号(med(irl))が、1(NaI)の時、

検出器中のエネルギー付与

ausgab の機能

- 検出器外部から、検出器に入射した各粒子のエネルギ情報の記録

```
!     ------------------------------------------------------
!     Score particle information if it enters from outside
!     ------------------------------------------------------
      if (irl .ne. irold .and. iarg .eq. 0) then
        if (iql .eq. 0) then              ! photon
          ie = e(np)/deltae +1
          if(ie .gt. 50) ie = 50
          spg(1,ie) = spg(1,ie) + wt(np)
        elseif (iql .eq. -1) then         ! electron
          ie = (e(np) - RM)/deltae +1
          if(ie .gt. 50) ie = 50
          spe(1,ie) = spe(1,ie) + wt(np)
        else                              ! positron
          ie = (e(np) - RM)/deltae +1
          if(ie .gt. 50) ie = 50
          spp(1,ie) = spp(1,ie) + wt(np)
        end if
      end if
    end if
```

粒子の移動に伴い、リージョンが変わる
=検出器の外から入射

howfarの役割

- howfar は、egs にジオメトリーに関する情報を伝えるサブルーチン
- howfar は、ustep の途中に、リージョン境界があるかどうかを調べる。ある場合には、
  - ustep を境界までの距離に置き換える
  - irnew を粒子が入っていくリージョン番号に設定する
- 粒子が、ユーザーが追跡を止めたい領域（例：体系外）に達したばあいには、idiscard フラグを1に設定する
- egs5では、ジオメトリー関連のルーチンがサブルーチンとして用意されている
- より詳細は、“ジオメトリーの書き方”の講義で説明