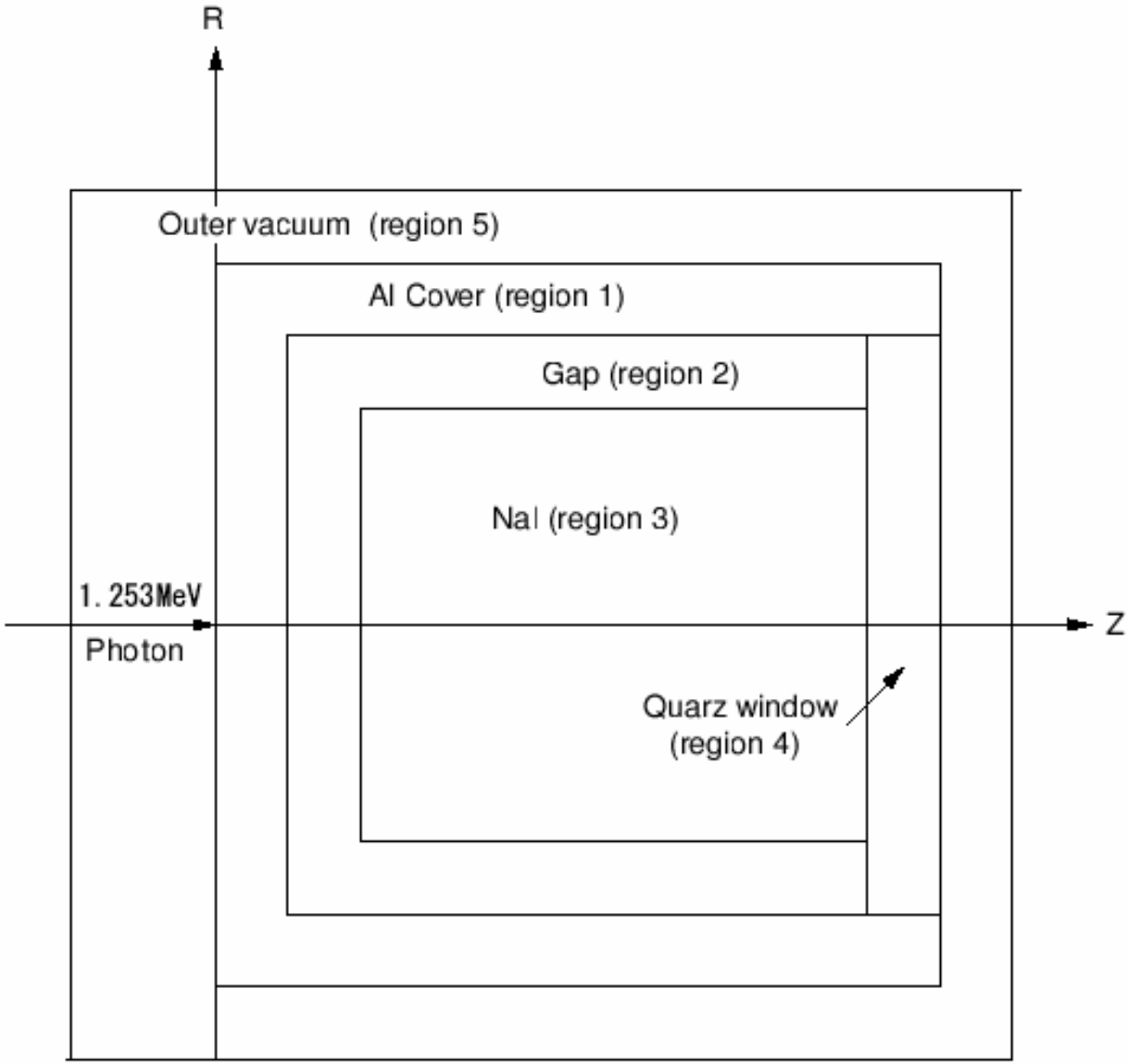
サンプルユーザーコード
ucnaicgv

平山　英夫、波戸　芳仁
KEK, 高エネルギー加速器研究機構

ucnaicgv.f

- 計算課題：NaI検出器のレスポンス計算
- 形状：CG形状(RCC：円筒)
- 1.253MeV γ線のペンシルビーム
- モードの選択（キーボード入力）
  - 飛跡表示モード(CGView)：egs5job.pic
  - 計算モード：egs5job.out


R
Outer vacuum (region 5)
Al Cover (region 1)
Gap (region 2)
NaI (region 3)
1.253MeV
Photon
Z
Quarz window
(region 4)

Step 1:Initialization

- egs5及びpegs5で使われているcommonは、それぞれincludeディレクトリー及びpegscommonsディレクトリーのファイルを　”include”文で取り込む
- 著者から提供されたジオメトリー関係などのユーザーコードのみで使用されるcommonは、auxcommonsディレクトリーのファイルをinclude文で取り込む

配列の大きさの指定

- commonで使用されている変数の配列の大きさは、parameter文で指定
  - egs5で使用されているcommonの変数は、include/egs5_h.f
  - ユーザーコードでのみ使用されるcommonの変数は、auxcommns/aux_h.f
- commonと同じようにinclude文により取り込まれる。
- 配列の大きさを変更する場合は、parameter文の変数を変更する

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

include 'include/egs5_bounds.f'
include 'include/egs5_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_thresh.f'
include 'include/egs5_uphiot.f'
include 'include/egs5_useful.f'
include 'include/egs5_usersc.f'
include 'include/egs5_userxt.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 'auxcommons/aux_h.f'   ! Auxiliary-code "header" file

include 'auxcommons/edata.f'
include 'auxcommons/etaly1.f'
include 'auxcommons/instuf.f'
include 'auxcommons/lines.f'
include 'auxcommons/nfac.f'
include 'auxcommons/watch.f'

include 'auxcommons/etaly2.f'      !  Added SJW for energy balance
```

ジオメトリー関係等ユーザーコードのみで使用されるcommon

CG関係のcommonで、CGを使用する場合には常に必要（変更無し）

```
include 'auxcommons/geom_common.f' ! geom-common file
 integer irinn
```

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)
```

このユーザーコード固有のcommon

main programで使用する倍精度の実数

```
 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
```

main programで使用する単精度の実数

main programで使用する整数

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

Step 2:pegs5-call

- 物質データ及び各物質のcharacteristic distance を設定した後で、pegs5をcallする。

```
nmed=3
medarr(1)='NAI                    '  pegs5で作成する物質データの
medarr(2)='AL                     '  名前。pegs5の入力データ(ユ
medarr(3)='QUARTZ                 '  ニット24から読み込み）と対応

do j=1,nmed
  do i=1,24
    media(i,j)=medarr(j)(i:i)
  end do
end do

chard(1) = 3.81d0      ! optional, but recommended to invoke
chard(2) = 0.1d0       ! automatic step-size control
chard(3) = 0.5d0
```

各物質のcharacteristic distance 当該物質のリージョンで中、最も小さいサイズを指定

Step 3:Pre-hatch-call-initialization

```
      npreci=2     ! Pict data mode for CGView

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrcin=0
      izonin=0
      itverr=0
      igmmax=0
      ifti = 4     ! Input unit number for cg-data
      ifto = 39    ! Output unit number for PICT

       write(39,100)
100   FORMAT('CSTA')
       call geomgt(ifti,ifto)
       write(39,110)
110   FORMAT('CEND')

!------------------------------
!     Get nreg from cg input data
!------------------------------
      nreg=izonin
```

CG関連の処理を行う部分。

CGを使用する場合は、変更しない。

```
RCC    1      0.00        0.0        0.0        0.00        0.0
              8.72        4.41
RCC    2      0.00        0.0        0.1        0.00        0.0
              8.12        4.31
RCC    3      0.00        0.0        0.6        0.00        0.0
              7.62        3.81
RCC    4      0.00        0.0        8.22       0.00        0.0
              0.5         4.31
RCC    5      0.00        0.0        -1.0       0.00        0.0
             10.0         5.00
END
Z1       +1      -2      -4
Z2       +2      -3
Z3       +3
Z4       +4
Z5       +5      -1
END
 2   0   1   3   0
```

Alカバーに内包される領域（円筒）を定義するためのbody

空隙に内包される領域（円筒）を定義するためのbody

NaIの領域（円筒）を定義するためのbody

フォトマルの窓ガラス領域（円筒）を定義するためのbody

体系全体を覆うbody

アルミニウム

空隙

NaI

窓ガラス

計算終了の領域

```
!   Set medium index for each region
!     Vacuum region
      med(nreg)=0
      med(2)=0      ! Inside vacuum
!     Al region
      med(1)=2

!     NaI detector region
      med(3)=1
      iedgfl(3)=1   ! 1:Produce flourscent X-rays
                    ! 0:Flourscent X-ray is not
produced

!     Quartz region
      med(4)=3

      do i=1,4
        if(i.ne.2) ecut(i)=0.561
      end do
!     ------------------------------
!     Set parameter estepe and estepe2
!     ------------------------------
      estepe=0.10
      estepe2=0.20
```

各リージョンへの物質、各種オプションの設定

リージョン3(NaI)で、特性X線の発生

エネルギーヒンジのためのパラメータ設定

estepe:最大エネルギーの電子・陽電子

estepe2:最小エネルギー電子・陽電子

リージョン毎に設定できるオプション

ecut, pcut	カットオフエネルギー (全エネルギー)
iphter	光電子の角度分布のサンプリング
iedgfl	K & L-特性X線の発生
iauger	K & L-オージェ電子の発生
iraylr	レイリー散乱
lpolar	光子散乱での直線偏光
incohr	S/Z rejection
iprofr	ドップラー広がり
impacr	電子衝突電離

乱数(ranlux乱数）

```
!     ------------------------------------------------------
!     Random number seeds.  Must be defined before call hatch
!     or defaults will be used.  inseed (1- 2^31)
!     ------------------------------------------------------
      luxlev = 1
      inseed=1
      write(1,150) inseed
150   FORMAT(/,' inseed=',I12,5X,
     *       ' (seed for generating unique sequences of Ranlux)')

!     =============
      call rluxinit  ! Initialize the Ranlux random-number generator
!     =============
```

異なったiseed毎に、重複しない乱数を発生することが可能

並列計算の場合に有効

Step 4: 入射粒子のパラメーター設定

```
iqin=0          ! Incident charge - photons
ekein=1.253       ! Kinetic energy
xin=0.0         ! Incident at origin
yin=0.0
zin=0.0
uin=0.0         ! Moving along z axis
vin=0.0
win=1.0
irin=1          ! Starts in region 2, could be 1
wtin=1.0        ! Weight = 1 since no variance reduction used

deltae=0.05       ! Energy bin of response
```

粒子の種類（電荷）

粒子の運動エネルギー

位置

方向

入射粒子のリージョン

線源粒子の位置から判断する場合は0に

```
      write(6,160)
160   format('Key in source type. 0:monodirectional, 1:point isotropic')
      read(5,*) isot
```

キーボード入力で、ペンシルビームか等方線源かの選択

Step 5:　hatch-call

- 電子･陽電子の全エネルギーの最大値をemaxeとして設定し、hatch を call する
- 読み込んだ情報を確認するために、物質データ及び各リージョンの情報を出力する

emaxe = ekein + RM　　! photon

線源粒子が光子の場合、近似的に線源光子のエネルギーに電子の静止エネルギーを加えた値を設定する

Step 6:Initialization-for-howfar

- ユーザーコードで使用する形状データを設定する
  - 平板、円筒、球などに関するデータ
- CGを使用しているこのユーザーコードでは、形状に関するデータは、cg入力データとしてstep 6以前に処理しているので、このstepで設定することはない

Step 7: Initialization-for-ausgab

- 計算で求める量の初期化、レスポンスのエネルギービン幅の設定等
- 計算したいヒストリー数(ncases)をキーボードからの入力で設定する
  - 飛跡表示モードの場合は、バッチ数(nbatch)も指定できるように（飛跡情報は、バッチ毎に、(ncases/nbatch)ケースずつ記録される
  - 計算モードの場合は、nbatchは、50

```
write(6,*) (' Key in number of cases.')
read(5,*) ncases
if(imode.eq.0) then
   write(6,*) (' Key in number of batch (=<50).')
   read(5,*)  nbatch
else
  nbatch = 50
end if
ncaspb = ncases / nbatch
```

Step 8: Shower-call

- 各バッチでncaspb数のヒストリーを、設定したバッチ数(nbatch:実験数に対応)だけ繰り返す
- 飛跡情報ファイルに、バッチ番号を記録する
- 各ヒストリー毎に、線源情報(粒子の種類、エネルギー、位置、方向）を設定

```
ekin = ekein
wtin = 1.0

wtsum = wtsum + wtin              ! Keep running sum of weights
etot = ekin + iabs(iqin)*RM       ! Incident total energy (MeV)
availke = etot + iqin*RM          ! Available K.E. (MeV) in system
totke = totke + availke           ! Keep running sum of KE
```

このユーザーコードでは、単一エネルギーの光子(iqin=0)なので、各ヒストリーで初期設定した同じ値を使用しているが、ヒストリー毎にエネルギーが異なる場合（分布している場合、複数のγ線を放出する線源）には、ekinを決定するサンプリングルーチンが必要

egs5で使用するエネルギー(showerに引き渡すエネルギー）は、全エネルギーなので、etotを設定する。（電子・陽電子の場合は、運動エネルギーに電子の静止質量を加える。

等方線源を選択した場合には、このルーティンで、線源の方向を決定する

WIを正(Zの正の方向のみ)に限定した場合の、rejection法による決定

```
      if (isot.eq.1) then            ! Sample isotropically.
380      call randomset(rnnow)
         zi0=rnnow
         call randomset(rnnow)
         xi0=2.0*rnnow-1.0
         call randomset(rnnow)
         yi0=2.0*rnnow-1.0
         rr0=dsqrt(xi0*xi0+yi0*yi0+zi0*zi0)
         if(rr0.gt.1.0) go to 380
         win = zi0/rr0
         uin = xi0/rr0
         vin = yi0/rr0
       end if
```

```
!---------------------------------------
!     Get source region from cg input data
!---------------------------------------
!
          if(irin.le.0.or.irin.gt.nreg) then
            call srzone(xin,yin,zin,iqin+2,0,irinn)
            call rstnxt(iqin+2,0,irinn)
          else
            irinn=irin
          end if
```

入射粒子の位置から、その場所のリージョン番号を求める

irin=0に設定しておけば、ここでリージョン番号が設定される

```
!         ============================================
          call shower (iqin,etot,xin,yin,zin,uin,vin,win,irinn,wtin)
!         ============================================
```

設定した条件でヒストリーを開始する

```
if (depe .gt. 0.D0) then
    ie=depe/deltae + 1
    if (ie .gt. 50)  ie = 50
    ph(ie)=ph(ie)+wtin
    tef=tef + wtin
    if(depe .ge. ekein*0.999) pef=pef +wtin
    depe = 0.D0
  end if
```

ヒストリー毎の情報を処理する。この処理が必要かどうかは、問題に依存する。

このユーザーコードでは、検出器の効率とレスポンスを計算することを目的としているので、吸収エネルギーが0でない場合は、当該ヒストリーでの検出器中の吸収エネルギーからエネルギー番号を決め、そこの値を+1する。吸収エネルギーの値から、全検出効率を+1し、吸収エネルギーが入射粒子の運動エネルギーと見なされる場合にはピーク検出効率を+1する。

この計算では、エネルギー吸収をあると全て検出されるとして全検出効率に加えているが、あるエネルギー以上のみを測定する結果との比較の場合には、ピーク検出効率と同じ様な判定が必要になる

Coincidence及びanti-coincidence

- 検出器間でcoincidenceやanti-coincidenceの計算を行う場合の、この例と同様にヒストリー終了毎に、処理を行う
  - Coincidenceの場合は、coincidenceをとる検出器の両方にエネルギー吸収があった場合にのみ主検出器の当該エネルギービンの値を+1増やす
  - Anti-coincidenceの場合は、逆に、主検出器以外の検出器にエネルギー吸収がない場合にのみ主検出器の当該エネルギービンの値を+1増やす

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

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - x)^2 \approx \overline{x^2} - (\bar{x})^2 ; (\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2)$$ x_i の分散

$$s_{\bar{x}}^2 = \frac{1}{N} s^2 \approx \frac{1}{N}[\overline{x^2} - \bar{x}^2]$$ $\bar{x}$ の分散

$$R = \frac{s_{\bar{x}}}{\bar{x}} \approx [\frac{1}{N}(\frac{\overline{x^2}}{\bar{x}^2} - 1)]^{1/2}$$ 相対標準偏差

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

$$\bar{x} = \frac{1}{n}\sum_{j=1}^{n} x_j$$ x_j の平均値

$$s_x^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - x)^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j^2 - \bar{x}^2)$$ x_j の分散

$$s_{\bar{x}}^2 = \frac{s_x^2}{n}$$ 平均の分散

$$FSD = \frac{s_{\bar{x}}}{\bar{x}}$$ 相対標準偏差

```
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

call plotxyz(99,0,0,0.D0,0.D0,0.D0,0.D0,0,0.D0)

write(39,410)          ! Set end of batch for CG View
410  FORMAT('9')
```

MORSE-CG の方法で誤差を評価するために、バッチ毎の平均値を計算する

実験毎の結果に対応

メモリーに残っている飛跡情報を飛跡情報ファイルに出力し、その後、バッチの終了を意味する'9'を書き込む

Step 9: Output-of-results

- 線源条件や、形状等の情報の出力
  - どの様な計算であるかを示すために出力
  - cgの場合は、形状をデータから直接示すことが容易でないので、必要な情報を設定して出力する
- バッチ毎の情報から、求めたい量の平均値と誤差(FSD)を計算し、出力する

ピーク検出効率

```
!     ---------------
!     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(1,470) avpe,sigpe
470   FORMAT(' Peak efficiency =',G15.5,'+-',G15.5,' %')
```

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を加算

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
- 使用するジオメトリールーティン毎に異なったhowfarとなる
  - cgを使用している場合は、このユーザーコードのhowfarを使用する

実習課題

- 実習課題１：NaI検出器の計算
  - 次のように変更して、ピーク検出効率及び全検出効率の変化を調べよ。
  - 1.253MeV線源について、一方向(Z-方向)のみに放出している線源光子を、等方線源に変更する。
  - 線源を、Cs-137の単一エネルギー光子(0.662MeV)に変える。
  - 線源を、Co-60に変え、1.173MeVと1.333MeV光子を同じ確率で発生させる。
  - 1.253MeV線源で、検出器の有感領域の厚さを2倍する。
- 実習課題２：Ge検出器の計算
  - 検出器を、Geに変更して、同じ大きさのNaIと、1.253MeV線源に対するピーク及び全検出効率と比較せよ。
- 実習課題３：空気電離箱の計算
  - 検出器を、20度C、1気圧の空気に変え、1.253MeV線源に対して、吸収エネルギーを求めよ。検出器の途中のギャップを除き、3インチ直径で3インチ長さの空気の領域の周辺に厚さ、5mmのAlがある形状とする。
  - 空気のW値(33.97eV/pair)を用いて、入射光子1個当たりのこの電離箱の出力(Coulomb/source)を求めよ。電荷素量を、1.602×10^{-19}C/e とする。