KEK Internal 99-5
July 1999
R/D

Lecture Notes of EGS4 Course at KEK (English Part)

(Last modified in 13 FEB 2003)

H. Hirayama and Y. Namito

High Energy Accelerator Reserach Organization

Lecture Notes of EGS4 Course at KEK
(English Part)

H. Hirayama and Y. Namito

High Energy Accelerator Reserach Organization
1-1 Oho, Tsukuba-shi, Ibaraki, 305-0801 Japan

Contents

Overview of the EGS4 Code System

1 The Main Features of The EGS4 Code System

The following is a summary of the main features of the EGS4 Code system, including statements about the physics that has been put into it and what can be realistically simulated.

- The radiation transport of electrons (+ or -) or photons can be simulated in any element, compound, or mixture. That is, a data preparation package, PEGS4, creates data to be used by EGS4, using cross section tables for elements 1 to 100.
- Both photons and charged particles are transported in random rather than in discrete steps.
- The dynamic range of the charged-particle kinetic energies goes from a few tens of keV up to a few thousand GeV. Conceivably, the upper limit can be extended higher, but the validity of the physics remains to be checked.
- The dynamic range of the photon energies lies between 1 keV and one thousand GeV (see above the statement).
- The following physics processes are taken into account by the EGS4 Code System:
  - Bremsstrahlung production (excluding the Elwert correction at low energies).
  - Positron annihilation in flight and the annihilation quanta are followed to completion.
  - Molière multiple scattering (*i.e.*, Coulomb scattering from nuclei). The reduced angle is sampled from a continuous (rather than discrete) distribution. This is done for arbitrary step sizes, selected randomly, provided that they are not so large or small as to invalidate the theory.
  - Møller (e^-e^-) and Bhabha (e^+e^-) scattering. Exact rather than asymptotic formula are used.

    Continuous energy loss applied to charged-particle tracks between discrete interactions.
    - The total stopping power consists of soft bremsstrahlung and collision loss terms.
    - The collision loss is determined by the (restricted) Bethe-Bloch stopping power with a Sternheimer treatment of the density effect.
  - Pair production.
  - Compton scattering.
  - Coherent (Rayleigh) scattering can be included by means of an option.
  - Photoelectric effect.
    - Neither fluorescent photons nor Auger electrons are produced or transported in the default version of the subroutine `PHOTO`.
    - Although other *user − written* versions of `PHOTO` can be created, they allow for the production and transport of K- and L-edge photons.
- PEGS4 is a stand-alone data preprocessing code consisting of 12 subroutines and 85 functions. The output is in a form for direct use by EGS4.
  - PEGS4 constructs piecewise-linear fits over a large number of energy intervals of the cross section and branching ratio data.

  - In general, the user needs only use PEGS4 *once* to obtain the media data files required by EGS4.
  - The PEGS4 control input uses the `NAMELIST` read facility of the `FORTRAN` language In Mortran3 form).
  - In addition to the option needed to produce data for EGS4, PEGS4 contains options to plot any of the physical quantities used by EGS4, as well as to compare sampled distributions produced by the `UCTESTSR` User Code with theoretical spectra.

- EGS4 is a package of subroutines plus block data with a flexible user interface. The division between the user-interface and EGS4 is shown in Figure 1.
  - This allows for greater flexibility without requiring one to be overly familiar with the internal details of the code.
  - Together with the macro facility capabilities of the Mortran3 language, this reduces the likelihood that user edits will introduce bugs into the code.
  - EGS4 uses the material cross section and branching ratio data created and fit by a companion code, PEGS4.

- The geometry for given problem is specified by a $user-written$ subroutine called `HOWFAR`, which, in turn, can make use of auxiliary subprograms.
  - Auxiliary geometry routines for planes, cylinders, cones, spheres, *etc.*, are provided with the EGS4 Code System for those who do not wish to write their own.
  - Macro versions of these routines are also provided in the set of defining macros (*i.e.*. in the `EGS4MAC` file) which, if used, generally result in a faster running simulation.
  - The `MORSE-CG` Combinatorial Geometry package can be incorporated into `HOWFAR` (*e.g.*, see the `UCSAMPCG` file on the EGS4 Distribution Tape). However, experience indicates that a much slower simulation generally results (on the order of at least a factor of four).
  - Transport can take place in a magnetic field by writing a specially designed `HOWFAR` subprogram, or in a more general matter (*e.g.*, including electric field) by making use of Mortran3 macro templates that have been appropriately placed for that purpose in the subroutine `ELECTR`.

- The user scores and outputs information in a $user-written$ subroutine, called `AUSGAB`.
  - The auxiliary subprogram `ECNSV1` is provided in order to keep track of energy for conservation (or other) purposes.
  - The auxiliary subprogram `NTALLY` is provided in order to keep the track of number of times energy has been scored into the `ECNSV1` arrays (*i.e.*, an event counter).
  - The auxiliary subprogram `WATCH` is provided in order to allow event-by-event or step-by-step tracking of the simulation.

- EGS4 allows for the implementation of *importancesampling* and other variance reduction techniques (*e.g.*, leading particle biasing, splitting, path length biasing, Russian roulette, *etc.*).

- Initiation of the radiation transport:
  - An option exits for initiating a shower with two photons from π_0 decay (*i.e.*, use `IQI=2` in the `CALL SHOWER` statement).

– The user has the choice of initiating transport by means of a monoenergetic particle, or by sampling from a known distribution (*e.g.*, a synchrotron radiation spectrum).

– Transport can also be initiated from a source having spatial and/or angular distributions.

2 The User Code and Method of Cascade Treatments in EGS4

As shown in Fig. 1, a user must provide the User Code to use the EGS4 system. The User code consists of at least from the main program, `SUBROUTINE AUSGAB` and `SUBROUTINE HOWFAR`.

The main program includes the following statements:

- to define parameters related to the geometry,
- to define the material data used and assign them to each region,
- to set initial data of a source particle,
- to call `SUBROUTINE HATCH` to read the material data calculated in PEGS4,
- to call `SUBROUTINE SHOWER`
- to analyze the results after finishing the required histories and output the analyzed results.

`AUSGAB` and `HOWFAR` must always be included in the User Code, even though they have no real function. (For example, `HOWFAR` including only `RETURN; END;` is provided for infinite geometry.)

`AUSGAB` is in default called in the following cases:

- A particle is going to be transported by distance `TVSTEP` (`IARG=0`).
- A particle is going to be discarded because its energy is below the cutoff `ECUT` (for charged particles) or `PCUT` (for photons) – but its energy is larger than the corresponding PEGS cutoff `AE` or `AP`, respectively (`IARG=1`).
- A particle is going to be discarded because its energy is below both `ECUT` and `AE` (or `PCUT` and `AP`) (`IARG=2`).
- A particle is going to be discarded because of a user request (in `HOWFAR` usually) (`IARG=3`).
- A photoelectron interaction has occurred and either:
  1. the energy of the incident photon was below the K-edge binding energy and it is going to be discarded, or
  2. a (fluorescent) photon is going to be discarded with the K-edge binding energy.

  (`IARG=4`)

The meanings of another `IARG` are given in Table 1.

In an electro-magnetic interaction, the number of particles becomes twice at each interaction. EGS4 treats this phenomenon as follows by using the stack number `NP`:

1. Set the stack number of the source particle to 1 (`NP=1;`).
2. After an interaction, the stack number `NP+1` or `NP` is assigned to a particle having a lower or higher total energy, respectively.
3. Follows the particle of the larger `NP` first.

4. If a particle is discarded for various reasons, it follows the particles of the stack number NP=NP-1 next.

5. A history is finished when the particle of NP=1 is discarded.

This is shown in Fig. 2 schematically.

3 The Meaning of the Main Variables used in EGS4

The variables used in EGS4 and their meanings are given in APPENDIX 2 of SLAC-265.

The main variables which are necessary to write the User Code are as follows:

COMMON/STACK	
X(NP), Y(NP), Z(NP)	Position of a particle.
U(NP), V(NP), W(NP)	Directional cosines of a particle.
WT(NP)	Statistical weight of the current particle (default=1.0).
E(NP)	Total energy in MeV.
IQ(NP)	Integer charge of a particle (+1, 0, -1).
IR(NP)	Index of a particle's current region.
NP	The stack pointer (*i.e.*, the particle currently being pointed to).
COMMON/BOUNDS	
ECUT	Array of regions' charged particle cutoff energies in MeV.
PCUT	Array of regions' photon cutoff energies in MeV.
COMMON/EPCONT	
EDEP	Energy deposited in MeV.
TSTEP	Distance to the next interaction(cm).
TVSTEP	Actual total (curved)step length to be tranported.
IROLD	Index of previous region.
IRNEW	Index of new region.
COMMON/MEDIA	
NMED	number of media being used (default=1).
RLC	Array containing radiation length of the media in cm.
RHO	Array containing the density of media in b/cm^3.

Initial parameters of source particles are defined as the following variables and transferred to SUBROUTINE SHOWER as the argument of SUBROUTINE CALL:

XI, YI, ZI	: Position of a source particle
UI, VI, WI	: Direction cosines of a source particle
EI	: Total energy of a source particle
IQI	: Integer charge of a source particle
IRI	: Index of a source particle's incident region
WTI	: Statistical weight of a source particle

4 Methods to Score the Various Information

In EGS4, the user must write SUBROUTINE AUSGAB depending on the information to be calculated. The following are simple explanations about the methods used to obtain the typical quantities to be calculated.

4.1 Calculation of the Absorbed Energy

The most common physical quantity is the absorbed energy. The variable EDEP gives the deposited energy. The treatment of this variable depends on the problems. For example, EDEP

inside the detector region is summed to calculate the detection efficiency or that for designing of the sampling calorimeter.

EDEP is summed at each region for the distribution calculation of the absorbed energy inside the material.

4.2 Calculation of the Particle Flux

The particle flux can be scored in the following way:

1. Select a particle that crosses the plane by (IF(IR(NP).NE.IROLD.AND.IARG.EQ.0)).
2. Select a particle depending on its charge.
3. Determine the energy and/or angular bin if the differential flux is necessary.

If the particle flux is at the specified plane, an additional condition to check the plane number crossed is required.

A correction due to the incident angle to the plane must be applied for calculating of the particle flux.

4.3 Calculation of the Track Length

The track length can be calculated by summing the actual total (curved) step length to be transported, TVSTEP. If the energy differential track length is required, the energy bin of the particle must be determined before the scoring process.

4.4 Way to Use Statistical Weight

In some cases, a statistical weight is assigned to a particle. The quantities multiplied by the statistical weight must be summed instead of the scored quantities (*e.g.*, EDEP*W(NP) instead of EDEP).

5 Array Sizes of the Variables Used in EGS4

The default array sizes of variables used in EGS4 are set to the following values at the EGS4MAC:

$MXMED — 10; Number of materials used in the User Code.
$MXREG — 2000; Region number.
$MXSTACK — 40; Maximum value of the stack pointer NP.

These values can be over-ridded in the User Code as shown in "How to Use Mortran3" in detail.

The default array sizes related to the geometry routines are also set to the following default values at EGS4MAC:

$MXPLNS — 100; Maximum number of planes.
$MXCYLS — 75; Maximum number of cylinders.
$MXSPHE — 75; Maximum number of spheres.
$MXCONES — 75; Maximum number of cones.

Table 1(a) Value for IARG and corresponding situation.

IARG	IAUSFL	Situation
0	1	A particle is going to be transported by distance TVSTEP.
1	2	A particle is going to discarded because its energy is below the cutoff ECUT (for charged particles) or PCUT (for photons)—but its energy is larger than the corresponding PEGS cutoff AE or AP, respectively.
2	3	A particle is going to be discarded because its energy is below both ECUT and AE (or PCUT and AP).
3	4	A particle is going to be discarded because a user requested it (in HOWFAR usually).
4	5	A photoelectric interaction has occurred and either: a) the energy of the incident photon was below the K-edge binding energy and it is going to be discarded, or b) a (fluorescent) photon is going to be discarded with the K-edge binding energy.

Table 1(b) Value for IARG and corresponding situation.

IARG	IAUSFL	Situation
5	6	A particle has been transported by distance TVSTEP.
6	7	A bremsstrahlung interaction is to occur and a call to BREMS is about to be made in ELECTR.
7	8	Return to ELECTR after a call to BREMS was made.
8	9	A Møller interaction is to occur and a call to MOLLER is about to be made in ELECTR.
9	10	Return to ELECTR after a call to MOLLER was made.
10	11	A Bhabha interaction is to occur and a call to BHABHA is about to be made in ELECTR.
11	12	Return to ELECTR after a call to BHABHA was made.
12	13	An in-flight annihilation of positron is to occur and a call to ANIHI is about to be made in ELECTR.
13	14	Return to ELECTR after a call to ANIHI was made.
14	15	A positron has annihilated at rest.
15	16	An pair production interaction is to occur and a call to PAIR is about to be made in PHOTON.
16	17	Return to PHOTON after a call to PAIR was made.
17	18	A Compton interaction is to occur and a call to COMPT is about to be made in PHOTON.
18	19	Return to PHOTON after a call to COMPT was made.
19	20	A photoelectric interaction is to occur and a call to PHOTO is about to be made in PHOTON.
20	21	Return to PHOTON after a call to PHOTO was made (assuming NP is non-zero).
21	22	Subroutine UPHI was just entered.
22	23	Subroutine UPHI was just exited.
23	24	A coherent (Rayleigh) interaction is about to occur.
24	25	A coherent (Rayleigh) interaction has just occured.

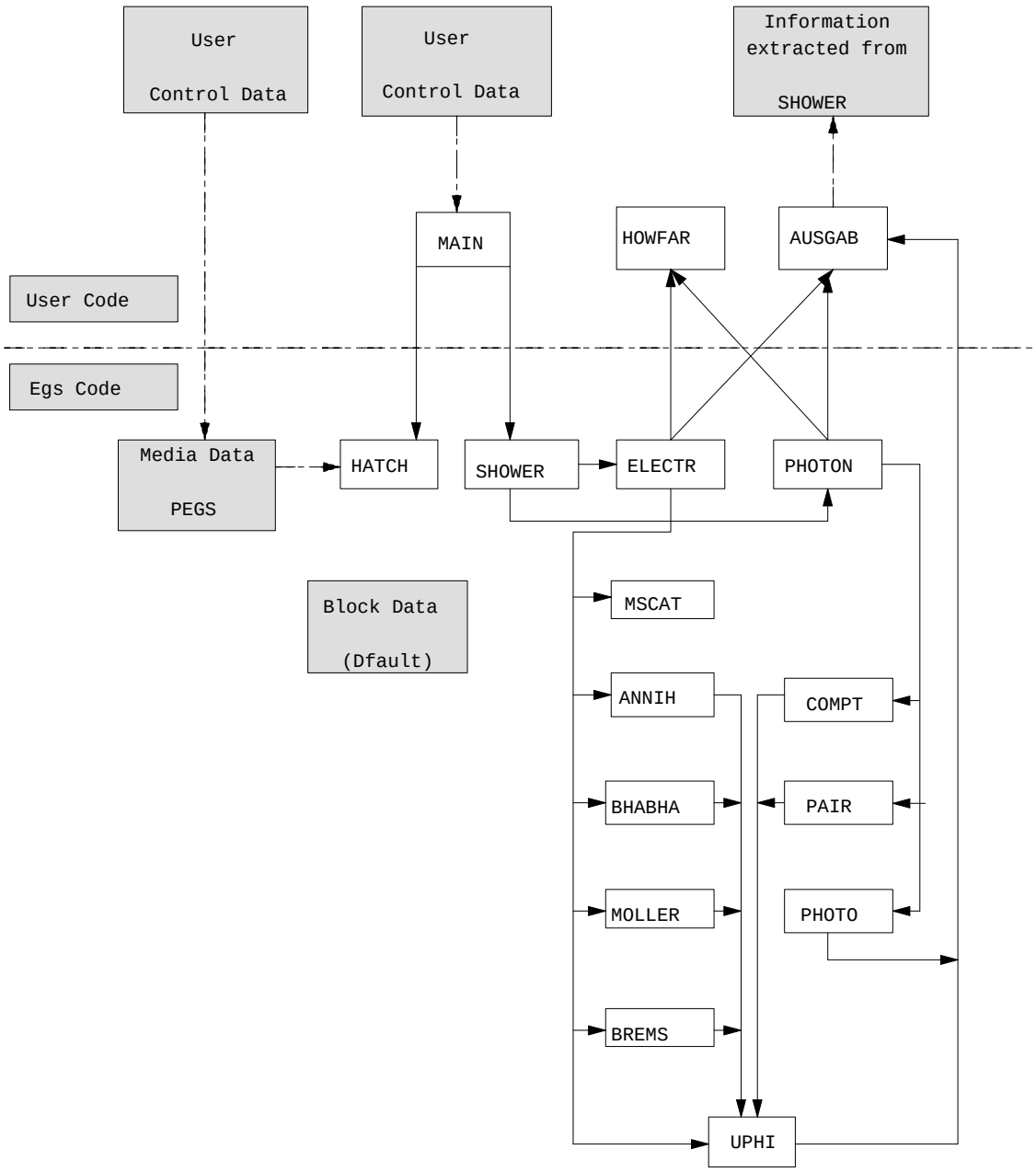


Figure 1: Division between the user-interface and EGS4.

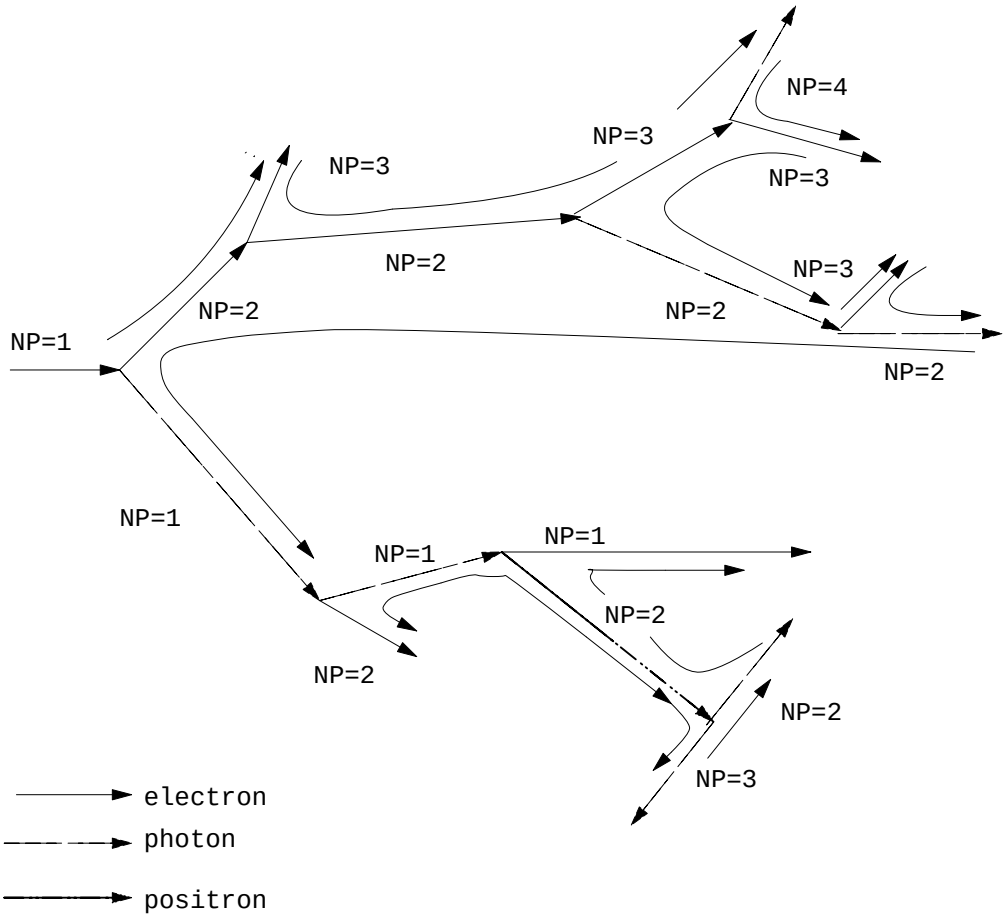


Figure 2: Flow control of cascade in EGS4.

How to Use Mortran3

1 Introduction

Mortran is a pre-processor of FORTRAN developed by J. Cook et al. at SLAC. EGS4[1] runs most effectively where written in Mortran3[2].

This lecture does not cover all of the rules of Mortan3. An elementary part will simply provide enough examples, including typical encountered errors, to illustrate the main features. An advanced part will show useful functions of macros together with the macros used in the EGS4 system.

2 Elementary Mortran3

2.1 Statement

Statement terminates with a semicolon (;).

Statements start in any column and continue to the next line without any special treatment. It is possible to write more than one statement on one line.

A beginner frequently forgets to put a semicolon at the end of a statement. If a semicolon is put at the end of each statement and a continuation character is deleted, FORTRAN statements can be recognized as Mortan statements.

2.2 Comment

A part enclosed by (") is treated as a comment in Mortran3.

A comment can be written any where in a program. It is also possible to write a long comment continuing over several lines. It is recommended to use comments in your user code to explain the meanings of variables *etc.* for yourself and for other users.

2.3 DO-loop

DO-loop is simplified ....just use brackets([and]), like DO I=J,K,N [....;].

It is not necessary to put a statement number and a "`CONTINUE`" statement. A function of J, K, N is the same one in a `DO-loop` of `FORTRAN`. Multiple use of a `DO-loop` is possible, like in the following example.

DO I=1,10 [DO J=1,20 [DO K=1,5 [A(I,J,K)=I*J*K;]]]

The number of "[" must be same as that of "]".

2.4 IF Statement

The `IF` statement is written like

```
IF e [.....;]
```

Here "e" is a logical expression with or without ( ), like in the following example.

```
IF A.LT.B [ C=D; E=F;] G=H; or IF(A.LT.B) [ C=D; E=F;] G=H;
```

`IF-ELSE` statements are written as follows, and may be nested to any depth:

```
IF e [.....;] ELSE [.....;]
```

"`THEN`" or "`ENDIF`", which is needed in `FORTRAN`, is not necessary.

`IF-ELSEIF` statements are written as

```
IF p [..;] ELSEIF q [....;] ELSEIF r [..;] ELSE [...;]
```

The number of "[" and "]" must be the same, as in the case of a `DO-loop`.

2.5 Input/Output

The standard Input/Output used in `FORTRAN [READ(5,` and `WRITE(6,]` can be expressed simply in Mortran3 as follows:

```
INPUT i-o list; (format list); corresponding to READ(5,
Example : INPUT NCASES;(I6);
OUTPUT i-o list; (format list); corresponding to WRITE(6,
Example : OUTPUT IXX;(' IXXST=',I12);
```

Although a statement number of the `FORMAT` is not necessary, a "format list" must be written to each `INPUT` or `OUTPUT`.

In the case of Input/Output to another `UNIT` than 5 or 6,

```
READ(1,:FORMAT1:) i-o list; :FORMAT1:FORMAT(format list);
WRITE(7,:FORMAT7:) i-o list;:FORMAT7:FORMAT(format list);
```

are used. `:FORMAT1:, :FORMAT7:` are examples of a label. Any labels are possible to use, of course. In this type of Input/Output, a `FORMAT` statement is not necessary to put after a `READ` or `WRITE` statement, and is set only once if the same format is used at many `READ` or `WRITE` statements.

2.6 Multiple Assignment

It is possible to assign a value to several variables with

/p,q,r,.....,z/=e;

The type of variables is not necessary same. For example,

```
/I, A(I,K), J/ = SQRT(X/2.0);
```

is converted to the following FORTRAN statements:

```
I = SQRT(X/2.0)
A(I,K) = SQRT(X/2.0)
J = SQRT(X/2.0)
```

Multiple assignments are useful as an initialization of variables before "`SHOWER CALL`".

2.7 Label

A part closed by ":" is recognized as a label. Example :USER-PHOTON-DISCARD:.....;

The number of characters of a label is unlimited. It is better to use a name which has an easily understood meaning. A label is replaced by a statement number in `FORTRAN`.

2.8 Iteration

There are several iterations which can be used in Mortran3.

WHILE e [.....;] LOOP [.....;] WHILE e; UNTIL e [.....;]

In the first example, the first logical "e" is checked. If "e" is true, statements inside the block are executed. If it is false, go to the next statement after the block.

In the case of a `LOOP`, logical "e" is checked after executing statements inside the block.

In the case of `UNTIL`, statements inside a block are executed until logical "e" becomes true.

In a similar way, the following iterations are also possible:

WHILE e [.....;] UNTIL f; WHILE e [.....;] WHILE f; UNTIL e [.....;] WHILE f; UNTIL e [.....;] UNTIL f;

`FOR-LOOP` can be used as follows:

FOR v=e TO f BY g [.....;] FOR v=e BY g TO f [.....;] FOR v=e TO f [.....;]

Here, "v" is a control variable and "e, f, g" are arbitrary numerical expressions. In the case of the last example, "g" is treated as "1".

As a special iteration, the following "Forever-Loops" are used:

```
LOOP [.....;] REPEAT;
LOOP [.....;]
```

The following statements with conditionals are used to get out of a "Forever-Loop".

```
GO TO :CHICAGO:;
EXIT;
IF (e) EXIT;
IF e [.....; EXIT;]
```

2.9 Operator

As relational operators, the following marks can be used in addition to `.LT.`, `.EQ.`, etc, which are used in `FORTRAN`.

```
<, <=, =<, =, ~=, =>, >=, >
```

As logical operators, the following symbols can also be used in addition to `.AND.`, `.OR.` etc.

```
<, < , & (and), | (or), ~ (not)
```

However, these symbols can not be mixed with an expression used in `FORTRAN` in the same statement.

For example:
`IF(A>B & C>D)` or `IF(A.GT.B.AND.C.GT.D)` is OK.....but
`IF(A>B.AND.C>D)` is not OK!

2.10 Techniques for the "Bracketing Out" Code

There are two ways to remove sections of code temporarily:

- Obvious way −− enclose between double-quote marks (").
- Another (undocumented) way is to use a `NONGENERATE` command.

2.10.1 `NONGENERATE` command

A `NONGENERATE` or `GENERATE` command is used as follows:

```
!NEWCONDITIONAL;  "Place at top of User Code"

NONGENERATE;   "Remove this code"
(lines of code)
ENDGENERATE;

GENERATE;       "Use this code"
(lines of code)
ENDGENERATE;
```

The parts of lines between NONGENERATE and ENDGENERATE are removed from the Mortran program.

This function can be used conditionally as follows:

```
$Select_Code_NEW;   "Use NEW or OLD code"

REPLACE{$Select_Code_#;} WITH {
   [IF] '{P1}'='NEW'  [SET <NewCode>=GENERATE;
                       SET <OldCode>=NONGENERATE;]
   [IF} '{P1}'='OLD'  [SET <NewCode>=NONGENERATE:
                       SET <OldCode>=GENERATE;]
}


<OldCode>
(lines of code)
ENDGENERATE;

<NewCode>
(lines of code)
ENDGENERATE
```

If $Select_Code_NEW is defined like in the above example, lines between <OldCode> and ENDGENERATE are removed from a Mortran program. If $Select_Code_OLD is defined instead, lines between <NewCode> and ENDGENERATE are removed.

2.11 Control Command for Mortran, FORTRAN Lists Output

There are many commands to control the outputs of Mortran or FORTRAN lists. The following are frequently used, and their functions are given. Although a command starting from % must be written from column 1, a command starting from ! and ending with ; can be written any place.

%E : Change to a new page in a Mortran listing.

%L : or !LIST; Output a Mortran listing.

%N : or !NOLIST; Do not output a Mortran listing.

%In : or !INDENT Mn; Indent n-spaces at each nest in a Mortran listing.

%F : Change to the FORTRAN mode. It is possible to use FORTRAN statements mixed with Mortran statements by using %F and %M.[1]

%M : Change to the Mortran mode.

!INDENT Fn; : Indent n-spaces at each nest in a FORTRAN listing.

!COMMENTS; : Output Mortran comments as FORTRAN comments in a FORTRAN listing.

!NOCOMMENTS; : Do not output Mortran comments as FORTRAN comments.

%An : Control command to output Mortran comments as comments in a FORTRAN listing.

- **n = 0** Do not output in a FORTRAN listing. Same with !NOANNOTATE;.
- **n = 1** Output Mortran comments as comments of FORTRAN from 2 to 80 column. Same with !ANNOTATE;.
- **n = 2** Output Mortran comments as comment of FORTRAN from column 40 to 80.

[1]There is a bug in the Mortran3 processor that causes statements preceding %F to be "eaten up". To avoid this, simply add a dummy statement, like " ....."; (or even a semicolon statement) immediately before each %F.

%Qn : Control command of comments in Mortran （default is 0）

n = 0 A comment must be enclosed with " at the beginning and end. Multi-line comments are allowed in this mode.

n = 1 At the end of a line, " is not necessary. A Mortran comment finishes at the end of the line. Therefore, multi-lines comments are not allowed.

%E is frequently put at the end of a `SUBROUTINE` to make it easy to see a Mortran listing.

If you put %N or !NOLIST; at the end of your user code, Mortran listings of `EGS4BLOK` and `EGS4` will not be printed out.

2.12 Errors related to Mortran and how to fix them

- Error due forgetting to put ";" at the end of a statement
  If you forget to put ";" at the end of a statement, 2 statements are recognized as one statement by Mortran3. This does not cause errors in Mortran, but a `FORTRAN` error occurs at that position.

- Extra or lack of] and "
  The number of [and] must be exactly the same. If the number is not the same, a Mortran error occurs. If the number of] is less than that of [, the error message "`UNCLOSED BLOCKS`" is printed out at the end of a Mortran listing. To find the place, check the nest level printed at the left side of the Mortran statement. This must be 0 at the end of a program unit, like `SUBROUTINE`. A positive value of this nest level means a lack of], and a negative value means an extra].

  To find this, it is useful to put %E at the end of each program unit. Find %E and check the nest level at that point. If a nest level is not 0, there is an extra or lack of] in this program unit.

  If you forget to put " at the end of a comment, the following statements are recognized as comments and a Mortran listing will not be printed out to the end. "`FATAL STRING OR STATEMENT TOO LONG`" is printed out in a Mortran listing. In this case, a " will be printed to the left of the nest level.

- Relation between the block character] and ;
  ; must be put at the end of a statement. Therefore, ";" after] like .....]; causes an error in Mortran.

3 Advanced Mortran

3.1 Macros

3.1.1 Definition of Macros

A macro[2] is expressed as follows:

REPLACE {pattern} WITH {replacement}

If the same expression with a "pattern" is used in a program written in Mortran, it will be translated to FORTRAN after replacing this part with a "replacement". As "replacement", both a numerical value or statements cab be used. Foe example, if following macro is defined,

```
REPLACE {ARRAYSIZE} WITH {50}

DIMENSION X(ARRAYSIZE);.........DO J=1,ARRAYSIZE [...]
```

will be the same as

```
DIMENSION X(50);.............DO J=1,50 [.....]
```

Blanks in a program are ignored during a comparison with a "pattern". Therefore, if

```
REPLACE {SIGMA(1)} WITH {SIGMA1}
```

is defined as a macro,

```
SIGMA (1) and SIGMA(1)
```

is transformed to "SIGMA1". On the other hand, blanks in a "pattern" have some meaning. For example, if

```
REPLACE {DUMP X;} WITH {OUTPUT X; (F10.2);}
```

is defined as a macro,

```
DUMP X;
```

Although it is transformed to `OUTPUT X; (F10.2);`, a transformation is not applied to

```
Y = DUMPX;
```

The argument of the pattern part is expressed as #, and is expressed as {P1}, {P2} in the replacement part.

{P1} is the first argument and {P2} the second one. For example, if

```
REPLACE {PLUS #;} WITH {{P1}={P1}+1;}
```

is defined as a macro,

```
PLUS A(I,J,K);
```

[2]What traditionally has been called a macro in Mortran2 was renamed a transformation rule in Mortran3.

becomes

```
A(J,J,K)=A(I,J,K)+1;
```

In macros of Mortran3, **uppercase letters and lowercase letters are treated as different characters**. Almost all macros in EGS4 are written in **uppercase letter**. Therefore, it is better to use a macro in an **uppercase letters**. If you find macros in a FORTRAN listing, check whether the macro is written in an uppercase letter or not.

In a Mortran program, the latest pattern is used if the same pattern is multiply defined in the same program. Therefore, if you re-define macros defined in EGS4MAC in your user code, your re-defined macro will be applied in EGS4.

By using this feature, we change EGS4 without changing EGS4, itself. Macros in the EGS4 system start from $ to show that they are macros.

3.2 Macros in EGS4MAC

3.2.1 PARAMETER Macros

A PARAMETER macro is defined in the EGS4MAC.MOR:

```
REPLACE {PARAMETER #=#;} WITH { REPLACE {{P1}} WITH {{P2}}}
```

In EGS4MAC.MOR is a string:

```
PARAMETER $MXMED=10;
```

If $MXMED appears, it should be replaced by 10.

It is possible to use macros inside a "replacement", as shown above.

A PARAMETER macro is useful for defining of arguments of DIMENSION. By changing value in a PARAMETER macro, it is possible change all arguments, as shown below.

```
PARAMETER $NBATCH=5;
PARAMETER $NDET=11;
....
....
COMMON/TOTALS/PHEI($NDET,150),TRLG($NDET,150),DELE;
....
```

3.2.2 COMIN Macro

Consider the following macro in EGS4MAC.MOR:

```
REPLACE {;COMIN/#,#/;} WITH {;COMIN/{P1}/;COMIN/{P2}/;}
```

Upon finding the string

```
;COMIN/BOUNDS,EPCONT,STACK/;
```

which could be in the User Code, or any part of the EGS4 Code System files, the following is produced:

```
;COMIN/BOUNDS/; COMIN/EPCONT,STACK/;
```

which becomes further expanded to

```
;COMIN/BOUNDS/; COMIN/EPCONT/; COMIN/STACK/;
```

Each COMIN is expanded by the macros that define the individual COMMONs. For Example, COMIN/BOUND/; expanded to COMMON/BOUNDS/ECUT($MXREG$),$PCUT$(MXREG),VACDST; with the following macro:

```
REPLACE {;COMIN/BOUNDS/;} WITH
   {;COMMON/BOUNDS/ECUT($MXREG),PCUT($MXREG),VACDST;}
```

Using COMIN macros in the User Code, it is possible to exchange information between the User Code and the EGS4 Code. The names of important COMMON BLOCK COMMON used in the EGS4 System are given in Table 1 together with the included variables and their meanings.

A COMIN macro is useful for defining the labeled COMMON used between the main program and SUBROUTINEs. If we define the labeled COMMON with a COMIN macro, you can increase, decrease or change variables including COMMON only at the definition of COMIN.

If we define COMIN/TOTALS/; as

```
REPLACE {;COMIN/TOTALS/;} WITH
  {;COMMON/TOTALS/PHEI($NDET,150),TRLG($NDET,150),DELE;}
PARAMETER $NDET=10;


;COMIN/BOUNDS,EPCONT,TOTALS,USEFUL/;
```

will be expanded to

```
COMMOM/BOUNDS/ECUT........
COMMON/EPCONT/EDEP........
COMMON/TOTALS/PHEI(10,150),TRLG(10,150),DELE
COMMON/USEFUL/MEDIUM.....
```

3.2.3 $LGN (List-Generator) Macro

There are a number of what we call *list-generator* macros.

The list-generator macro

```
$LGN(A,B,C(123));
```

produces the string

```
A(123),B(123),C(123)
```

$LGN is often used in Block Commons. For example,

```
;COMMON/STACK/$LGN(E,X,Y,Z,U,V,W,DNEAR,WT,IQ,IR($MXSTACK)),NP;
```

ends up becoming the following FORTRAN statements:

```
      COMMON/STACK/E(40),X(40),Y(40),Z(40),U(40),V(40),W(40),DNEAR(40),W
     *T(40),IQ(40),NP
```

3.2.4 $RANDOMSET Macro

The purpose of a $RANDOMSET macro is to insert *in-line* code for a pseudo-random number generator for speeding up generation.

$RANDOMSET macro is used with COMIN/RANDOM/;. In the case of the main-frame computers, this macro is

```
REPLACE {;COMIN/RANDOM;} WITH {;COMMON/RANDOM/IXX; INTEGER IX(2);
REAL*8 DRN; EQUIVALENCE(IX(1),DRN); DATA IX(1)/Z46000000/;}
REPLACE {$RANDOMSET #;} WITH {IXX=IXX*663608941;IX(2)=IXX;
{P1}=DRN+0.0D0;}
```

On the other hand,

```
REPLACE {;COMIN/RANDOM;} WITH {;COMMON/RANDOM/IXX;}
REPLACE {$RANDOMSET #;} WITH {IXX=IXX*663608941;
   {P1}=0.5+IXX*0.23283064E-09;}
```

is used for UNIX, VAX and PC.

$RANDOMSET is used in the following example:

```
$RANDOMSET RN;  "Sample RN uniformly on (0,1)"
PHI=TWOPI*RN;   "Obtain azimuthal angle"
```

which leads to the following *in-line* FORTRAN (in the case of UNIX, etc.):

```
IXX=IXX*663608941
RN=0.5+IXX*0.23283064E-9
PHI=TWOPI*RN
```

The above random number generator is called RAN6, which comes as a default with the EGS4 distribution.

The Marsaglia-Zaman random number generator[3] is now being distributed. It has a long periodicity ($2^{144} \sim 10^{43}$), and is portable to all 32-bit machines.

3.2.5 Macros for Debugging Purposes

- $TRACE #; $S1TRACE #; $S2TRACE #; $ITRACE #; $IS1TRACE #; $IS2TRACE #;
  These macros are used to print out variables each time when they are calculated for debugging purposes to find the reasons for errors.

  $TRACE: Real-type variable without an argument

  $S1TRACE: Real-type variables with one argument

  $S2TRACE: Real-type variables with otwo arguments

  $ITRACE: Integer-type variables without an argument

  $IS1TRACE: Integer-type variables with one argument

  $IS2TRACE: Integer-type variables with two arguments

  $S2TRACE#; $ITRACE #; $IS1TRACE #; $IS2TRACE #; are created at KEK modifing $TRACE #; $S1TRACE # and included in kek4mac.mot.

  By adding DEBUG in COMIN of the main program and inserting QDEBUG=.TRUE.; at the beginning of an executable statement, these macros work.

- $CALLTRACE;
  If $CALLTRACE; is used instead of $TRACE etc. or together with them, the name of a SUBROUTINE is printed out each time it is CALLED. This is not applied to CALL just after "label".

3.2.6 SPECIFY and $SETINTERVAL#,#;

A specifier is the name of a recognition function and "the function" is defined by a SPECIFY statement:

```
SPECIFY specifier AS specification;
```

The following SPECIFY macro is in EGS4MAC:

```
SPECIFY SNAME AS ['SINC'|'BLC'|'RTHR'|'RTHRI'];
```

The specifier SNAME is defined for use within the pattern part of macros. This is demonstrated in the following $EVALUATE macro:

```
REPLACE {$SETINTERVAL#,#;}
         WITH {[IF] '{P2}'=SNAME [L{P1}={P2}1*{P1}+{P2}0;]
          [ELSE] [L{P1}={P2}1(MEDIUM)*{P1}+{P2}0(MEDIUM);]}
```

MEDIUM is the number of media to be calculated.

3.2.7 $EVALUATE Macro

One of the two forms of the $EVALUATE macro is:

```
REPLACE {$EVALUATE#USING#(#);} WITH {[IF] '{P2}'='SIN'
[{P1}={P2}1(L{P3})*{P3}+{P2}0(L{P3});] [ELSE]
[{P1}={P2}1(L{P3},MEDIUM)*{P3}+{P2}0(L{P3},MEDIUM);]}
```

Consider the following macros:

```
$RANDOMSET RNNO38;
PHI=RNNO38*TWOPAI;
$SET INTERVAL PHI,SINC;
$EVALUATE SINPHI USING SIN(PHI);
```

The second and forth are expanded into:

```
LPHI=SINC1*PHI + SINC0;
SINPHI=SIN1(LPHAI)*PHI + SIN0(LPHI);
```

LPHI is an integer corresponding to an interval in a piecewise-linear fit of a tabulated sine function, and SINPHI makes use of both PHI and LPHI in a table lockup.

3.2.8 $TRANSFERPROPERTIESTO#FROM#;

This macro is used to set parameters, like the position, weight etc., to a produced new particle. It is written as:

```
REPLACE {$TRANSFERPROPERTIESTO#FROM#;} WITH
   {X{P1}=X{P2};Y{P1}=Y(P2};Z({P1}=Z{P2};IR{P1}=IR{P2};
    WT{P1}=WT{P2};DNEAR{P1}=DNEAR{P2};}
```

For example,

```
$TRANSFERPROPERTIES TO (1) FROM I;
```

which is used at the top of SUBROUTINE SHOWER, produces:

```
X(1)=XI;  Y(1)=YI; Z(1)=ZI; IR(1)=IRI; WT(1)=WTI; DNEAR(1)=DNEARI;
```

3.2.9 Macros Inserted for the Future Including New Features by the User

- $KERMA-INSERT;: Used to calculate KERMA.DEFAULT is NULL.
- $DE-FLUCTUATION;: Used to include energy-loss model like "Landau Fluctuation". DEFAULT is NULL.
- $POSITRON-ECUT-DISCARD;: Macro to control the production of annihilation photons when a positron is DISCARDED in the energy region AE $<$ E $<$ ECUT. DEFAULT is used to produce annihilation photons with the following definition:

  ```
  REPLACE {$POSITRON-ECUT-DISCARD;} WITH {EDEP=PEIE-PRM;}
  ```

  If EDEP=PEIE+PRM is defined as $POSITRON-ECUT-DISCARD;, the production of annihilation photons is disabled.

- $PARTICLE-SELECTION-ELECTRA; $PARTICLE-SELECTION-PHOTON; etc.
  These macros are used to apply variance reduction techniques, like "Leading Particle" or "Particle Splitting", and are defined regarding all reactions for photons, electrons and positrons. (For Example, $PARTICLE-SELECTION-COMPT;) DEFAULT is NULL.

- $SELECT-ELECTRON-MFP;: Macro to determine the mfp of an electron. DEFAULT is;

```
REPLACE {$SELECT-ELECTRON-MFP;} WITH {$RANDOMSET RNNE1;
IF(RNNE1.EQ.0.0) [RNNE1=1.E-30;]
DEMFP=AMAX1(-ALOG(RNNE1),$EPSEMFP);}
```

- $SELECT-PHOTON-MFP;: Macro to determine the mfp of a photon
  DEFAULT is;

```
REPLACE {$SELECT-PHOTON-MFP;} WITH {$RANDOMSET RNNE1;
   IF(RNNO35.EQ.0.0) [RNNO35=1.E-30;]
   DEMFP=-ALOG(RNNO35);}
```

  In the case of applying variance reduction techniques, like an "Exponential Transform", this macro is modified.

- $USER-RANGE-DISCARD;: Macro to DISCARD an electron or positron if its range is smaller than the distance to the boundary to increase the computational efficiency. DEFAULT is NULL.

- $SET-TUSTEP-EM-FIELD; $SET-USTEP-EM-FIELD; $VACUUM-TRANSPORT-EM-FIELD; $SET-ANGLES-EM-FIELD; $SET-TVSTEP-EM-FIELD; $ADD-WORK-EM-FIELD;
  Macros to include a transport of charged particles inside a magnetic field. DEFAULT is NULL.

4 Examples of Application of EGS4MAC in User Code

4.1 Parameters used in EGS4

In each SUBROUTINE of EGS4, various parameters are calculated using those values calculated in PEGS4. These parameters are also possible to use in User Code. 、$SET INTERVAL and $EVALUATE macros are used in this case, as shown in the following example:

```
EIG=EI;
$SET INTERVAL GLE,EIG;"SET PWLF INTERVAL"
MEDIUM=MED(2);
GLE=ALOG(EIG);
$EVALUATE GMFPRO USING GNFP(GLE);
```

The mfp of a photon inside the medium of region number 2 is calculated with these macros. A variable "MEDIUM" is used in EGS4. Therefore, it is better to set "MEDIUM" to the one currently being used in the following way:

```
EIG=E(NP);
MEDOLD=MEDIUM;
MEDIUM=MED(2);
.......
MEDIUM=MEDOLD;
```

4.2 NEGATIVE-USTEP

It is possible to over-ride $USER-CONTROLS-NEGATIVE-USTEP; macro to print out more information than DEFAULT.

```
REPLACE{$USER-CONTROLS -NEGATIVE-USTEP;} WITH
  {;IF(USTEP.LT.-1.E-4) [OUTPUT USTEP;(' NEGATIVE USTEP=',G20.10);
  OUTPUT NP,IR(NP),IQ(NP);(' NP,IR,IQ=',3I5);
  OUTPUT X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP);
  (' X,Y,Z,U,V,W=',6G10.4); STOP;]}
```

3-5 lines are added to the DEFAULT one.

4.3 Examples of Uses of LATCH

It is possible to add a new property, except for the particle type, its position, direction and weight using "LATCH", which is included in COMIN/STACK/;.

- If you want to keep the type of particle when it enters into the specified region (for example, you want to calculate the energy deposition depending on the type of particle entering the detector (region 4).), LATCH is used at SUBROUTINE AUSGAB as follows:

```
IRL=IR(NP);
IF(IRL.NE.IROLD.AND.IRL.EQ.4) [LATCH(NP)=IQ(NP);
                    "Particle enters into detector"]
IF(LATCH(NP).EQ.0) [EPDE(IRL)=EPDE(IRL)+EDEP*DPWT;]
ELSE [EEDE(IRL)=EEDE(IRL)+EDEP*DPWT;]
```

  At the second Statement, it is checked whether the particle enters the region 4 or not. If it enters the detector region, LATCH(NP); is replaced with the type of entering particle.

- It is possible to score the energy deposition depending on the number of Compton scattering. Include COMIN/EPCONT to the COMIN of main program and set IAUSFL(18)=1; to CALL AUSGAB(IARG); when Compton scattering occurs. Write the following statement at the top of AUSGAB.

```
IF(IARG.EQ.17) [LATCH(NP)=LATCH(NP)+1; RETURN;]
```

  Now LATCH keeps the number of Compton scattering from the start of the recorded history. Score the energy deposition for the following case:

  No Compton : LATCH(NP)=0;

  Single Compton : LATCH(NP)=1;

  Multiple Compton : LATCH(NP)>1;

4.4 Add New Property to Particle

If you want to use new properties, such as the total time after stating a history and total track length, modify COMIN/STACK/ and $TRANSFERPROPERTIES macro as follow:

```
REPLACE {;COMIN/STACK/;} WITH
  {;COMMON/STACK/$LGN(E,X,Y,Z,U,V,W,DNEAR,WT,TTLSS,TIMESS,IQ,
   IR($MXSTACK)),NP; $ENERGY PRECISION E;}
REPLACE {$TRANSFERPROPERTIESTO#FROM#;} WITH
  {X{P1}=X{P2};Y{P1}=Y{P2};Z{P1}=Z{P2};IR{P1}=IR{P2};
  WT{P1}=WT{P2};DNEAR{P1}=DNEAR{P2};
  [IF] '{P2}'='I' [TTLSS(1)=0.0; TIMESS(1)=0.0;]
  [ELSE] [TTLSS{P1}=TTLSS{P2};TIMESS{P1}=TIMESS{P2};]}
```

Here, TTLSE is the total track length and TIMES is the time in nsec after starting a history. They are calculated using the following statements:

```
TTLSS(NP)=TTLSS(NP)+TVSTEP;
IF(IQ(NP).EQ.0) [ SPEED=2.998e+1;]
ELSE [SPEED=SQRT(BETA2)*2.998E+01;
                  "Speed of charged particle in cm/nsec"]
TIMESS(NP)=TIMESS(NP)+TVSTEP/SPEED;
```

STACK and EPCONT must be included in the COMIN of SUBROUTINE AUSGAB.

4.5 Errors related to Mortran or EGS4 execution and how to fix them

4.5.1 In the case when you cannot find reason of error in Mortran

Put `!TRACE 2;` and `!TRACE 1` at the statement several lines before and after any statement that causes an error, respectively. A detailed transformation will be printed out if macros exits between `!TRACE 2` and `!TRACE 1`. You may find the reason of an error by checking this output.

For example, if you put `!TRACE 2` and `!TRACE 1` as follows:

```
!TRACE 2;
$SET INTERVAL GLE,GE;
$EVALUATE GMFPR1 USING GMFP(GLE);
!TRACE 1;
```

The Mortran listing will be

```
        0  !TRACE 2;
REDUCED ;
        0  $SET INTERVAL GLE,GE;
MATCHED $SETINTERVALp,p;
  ARG 1 GLE
  ARG 2 GE
REDUCED LGLE=GE1(MEDIUM)*GLE+GE0(MEDIUM);
        0  $EVALUATE GMFPR1 USING GMFP(GLE);
MATCHED $EVALUATEpUSINGp(p);
  ARG 1 GMFPR1
  ARG 2 GMFP
  ARG 3 GLE
REDUCED GMFPR1=GMFP1(LGLE,MEDIUM)*GLE+GMFP0(LGLE,MEDIUM);
        0  !TRACE 1;
```

4.5.2 Large negative "USTEP"

If `USTEP` is negative and its absolute value is larger than $1.0X10^{-4}$, the "`NEGATIVE USTEP`" massage will be printed out. After that, `EGS4` will assign 0 to `USETP` and continue the calculation. If the absolute value of `NEGATIVE USTEP` is larger than 10^{-2}, you must check your geometry routine. (A small negative "`USTEP`" will occur due to the limited precision of the computers.)

4.5.3 Error related to the Subroutine `HATCH`

If your User Code stops while printing following message, check that the names of materials used in your user code are the same as those in material data calculated by `PEGS4`.

```
 END OF FILE ON UNIT 12
 PROGRAM STOPPED IN HATCH BECAUSE THE
 FOLLOWING NAME WERE NOT RECOGNIZED:
 ...........


 STOPPED IN HATCH:REQUESTED RAYLEIGH OPTION FOR MEDIUM ##
 BUT RAYLEIGH DATA NOT INCLUDED IN DATA CREATED BY PEGS
```

This message means that you will try to apply `RAYLEIGH OPTION` to the material of number ##, which does not include `COHERENT` data. You must make change so as not to apply `RAYLEIGH OPTION` to this material or re-create material data with IRAYL=1.

4.5.4 DIMENSION overflow

If a `DIMENSION` overflow occurs without any programming problem in your user code, check if the energy of the incident particle is really less than the upper energy of the material data. It is better to check the source energy in the User Code before `CALLING SHOWER`.

4.5.5 Find the reason of LOOP

(a) Insert `OUTPUT IXX;(' IXXST=',I12);` before `CALL SHOWER` and run program.

(b) Change `IXXST` to the last `IXX` and set history number to 1.
Define `$TRACE`, `$S1TRACE` and `$S2TRACE` (`$ITRACE, $IS1TRACE` and $IS2TRACE for integer) macros to check the positions, energy, direction, region number etc. Add `DEBUG` in `COMIN` of the main program.
Insert `QDEBUG=.TRUE.;` at the beginning of an executable statement.
Run the program.

(c) Check the results to find the reason.

References

[1] W. R. Nelson, H. Hirayama and D. W. O. Rogers, "THE EGS4 CODE SYSTEM", SLAC-265, December 1985.

[2] A.J.Cook,"Mortran3 User's Guide", SLAC Computational Research Group Technical Memorandom Number CGTM 209 (1983).

[3] G.Masaglia and A.Zaman,"A New Class of Random Number Generator", *Annals of Applied Probability* **1**, 462-480(1991).

Table 1. Names of the COMMON blocks important to the user with a brief description of their functions.

COMMON BLOCK	VARIABLE	FUNCTION
BOUNDS	ECUT	Array of regions' charged particle cutoff energies in MeV.
	PCUT	Array of regions' photon cutoff energies in MeV.
	VACDST	Distance to transport in vacuum (default=1.E8).
EPCONT	EDEP	Energy deposited in MeV (Double Precision).
	TSTEP	Distance to next interaction (cm)
	TUSTEP	Total (curved) step length requested.
	USTEP	User (straight line) step length requested and granted.
	TVSTEP	Actual total (curved) step length to be transported.
	VSTEP	Actual (straight line) step length to be transported.
	IDISC	User discard request flag (to be set in HOWFAR. IDISC $>$ 0 means user requests immediate discard, IDISC $<$ 0 means user requests discard after completion of transport, and IDISC=0 (default) means no user discard requested.
	IROLD	Index of previous region.
	IRNEW	Index of new region.
	RHOF	Value of density correction (default=1)
	EOLD	Charged particle (total) energy at beginning of step in MeV.
	ENEW	Charged particle (total) energy at end of step in MeV.
	BETA2	Beta squared for present particle.
	IAUSFL	Array of flags for turning on various calls to AUSGAB.
	EKE	Kinetic energy of charged particle in MeV.
	ELKE	Natural logarithm of EKE.
	GLE	Natural logarithm of photon energy.
	TSCAT	See Eq.2.14.82 in SLAC-265.
MEDIA	NMED	Number of media being used (default=1).
	MEDIA	Array containing names of media (default is NaI).
	IRAYLM	Array of flags for turning on (=1) coherent (Rayleigh) scattering in various media. Set in HATCH based on values of IRAYLR.
	RLC	Array containing radiation length of the media in cm.
	RLDU	Array containing radiation length of the media in distance units established by DUNIT.
	RHO	Array containing density of the media in g/cm^3.
MISC	MED	Array containing medium index for each region.
	DUNIT	The distance unit to be used: DUNIT=1 (default) establishes all distances in cm; whereas, DUNIT=2.54 establishes all distances in inches.
	KMPI	FORTRAN unit number (default=12) from which to read material data.
	KMPOI	FORTRAN unit number (default=8) on which to "echo" material data (e.g., printed output, "dummy" output, etc.).

Table 1. Names of the COMMON blocks important to the user with a brief description of their functions (continued).

COMMON BLOCK	VARIABLE	FUNCTION
MISC	RHOR	Array containing the density for each region (g/cm^3). If this is different than the default density for the medium for that region, the cross section and stopping power (with the exception of the density effect) are scaled appropriately.
	NOSCAT	Number of times multiple scattering has been bypassed in subroutine MSCAT (initialized to 0 in BLOCK DATA).
	IRAYLR	Array of flags for turning on (=1) coherent (Rayleigh) scattering in various regions (default=0).
RANDOM	IXX	Random number generator seed (default=123456789).
STACK	note:	This COMMON block contains the information about the particles currently in the shower. All of the following variables are arrays except NP.
	E	Total energy in MEV (Double precision).
	X, Y, Z	Position of particle in units established by DUNIT.
	U, V, W	Direction cosines of particle (not necessarily normalized −− see Section A2.4.1c of SLAC-265).
	DNEAR	A lower bound of distance from (X,Y,Z) to nearest surface of current region.
	WT	Statistical weight of current particles (default=1.0), To be used in conjunction with variance reduction techniques as determined by user.
	IQ	Integer charge of particle (+1, 0, -1).
	IR	Index of particle's current region.
	NP	The stack pointer (.i.e., the particle currently pointed to. Also, the number of particles on the stack.
THRESH	RMt2	Twice the electron rest mass energy in MeV.
	RMSQ	Electron rest mass energy squared in MeV-squared.
	AP	Array containing PEGS lower photon cutoff energy for each medium in MeV.
	UP	Array containing PEGS upper photon cutoff energy for each medium in MeV.
	AE	Array containing PEGS lower charged particle cutoff energy for each medium in MeV.
	UE	Array containing PEGS upper charged particle cutoff energy for each medium in MeV.
	TE	Same as AE except kinetic energy rather than total energy.
	THMOLL	Array containing the Moller threshold energy (THMOLL=AE+TE) for each medium in MeV.

Table 1. Names of the COMMON blocks important to the user with a brief description of their functions (continued).

COMMON BLOCK	VARIABLE	FUNCTION
UPHIOT	THETA	Collision scattering angle (polar).
	SINTHE	Sine of THETA.
	COSTHE	Cosine of THETA.
	SINPHI	Sine of PHI (the azimuthal scattering angle of the collision).
	PI	Pi.
	TWOPI	Twice pi.
USEFUL	MEDIUM	Index of current medium. If vacuum, then MEDIUM=0.
	MEDOLD	Index of previous medium.
	RM	Electron rest mass energy in MeV.
	PRM	"Precision" electron rest mass energy in MeV (Double precision).
	PRMT2	Twice PRM (Double precision).
	IBLOBE	Flag indicating if photon is below binding energy (EBINDA) after a photoelectric interaction 8yes=1).

How to use PEGS4

PEGS4 is a program used to create material data to be used in EGS4. The user must create an input file to run PEGS4. The input file is categorized as "element", "compound" or "mixture". In this manual, the way to create an input file for these 3 categories is described. This manual is specific for several examples, and detailed descriptions are given concerning points about which users tend to make errors. On the other hand, a general manual for PEGS4 is available as A3.3 of SLAC-265 "PEGS4 Options and Input Specifications". It is recommended to read both this manual and A3.3 of SLAC-265.

1 Input file for element

For example, an input file used to create material data of iron between 10 keV and 50 MeV is:

```
ELEM
 &INP IAPRIM=1,IRAYL=1 &END
FE-RAYLEIGH                    FE
FE
ENER
 &INP AE=0.521,UE=50.511,AP=0.01,UP=50.0 &END
TEST
 &INP &END
PWLF
 &INP &END
DECK
 &INP &END
```

- "ELEM" in the first line indicates that the material is composed of a single element. (Input from the 1st column.)
- Lines 2, 6, 8, 10 and 12 are "namelist input". This input format is IBM's fortran expansion. [1]
- "IAPRIM=1" is an option used to normalize the bremsstrahlung cross section so that PEGS4 gives the same radiative stopping power for electrons as given in ICRU Report 37. [2] This option should always be used.
- "IRAYL=1" is an option used to include Rayleigh scattering (elastic and coherent scattering of photons by an atomic electron).
- "FE-RAYLEIGH" is a material data's name. This name ise used when material data are specified for each region in the EGS4 user's code. This name must be equal to or less than 24 letters; a user can freely select the name. In the case that the material name is shorter than 24 letters, a space of "24 – material data name length" should be added in the user's code.
- From the 31st column in the 3rd line, "FE" specifies the corresponding Sternheimer-Seltzer-Berger coefficient ("SSB" coefficient) for a density effect.
  - If the target material is involved in Table 2.13.2 of SLAC-265, input the name in that table (from the 31st column!)

[1] In "namelist input", default values for the variables are usually written in the program, and values are overwritten by input when necessary. In namelist input, sets of "Variable name=value" are written between &INP and &END while separating them by "," . Plural line input is capable in namelist input. In the case of MS-FORTRAN, "/" is used instead of "&END".

[2] One expansion at NRCC. Thus, no description was made in SLAC-265.

  - If the target material is not involved in that table, this input is ignored, and the density effect is calculated using Sternheimer-Peierls' general formula

- "FE" in the 4th line is the atomic symbol for the element. PEGS4 determines the element by this symbol, and sets the parameters (ex. density) automatically.

- "ENER" specifies starting the input of the energy range of the material data. The energy range is input in the 6th-line.

  - AE and UE are the lower and upper boundary energy of the material data for an electron and a positron (in unit of MeV). AE and UE are specified in the "TOTAL energy", including the rest-mass energy.
    * If 'AE<0.511' is input, it means the specification of negative kinetic energy. PEGS4 operates in an irregular condition and outputs confused data. In this case, much time is consumed to find the reason. it is thus advisable to be careful not to input data like this.
    * If 'UE=50.0' is input, material data up to a kinetic energy of 49.489 MeV are created. If an electron of kinetic energy of 50 MeV is specified as the incident particle in EGS4, that causes an irregular operation of EGS4 due to access to outside the energy region of the material data. To prevent this, be careful to add the electron rest mass energy when UE is specified.
  - AP and UP are the lower and upper boundary energy of the material data for a photon (in MeV unit).

  How should the energy range be specified? If an energy range that is too wide is specified, the material data become large or the precision may be slightly little worse. On the other hand, if the same energy range as that used in the EGS4 simulation is specified in PEGS4, it is not convenient because material data must be created using PEGS4 each time the energy range of the simulation is changed. Thus, it is best to create material data in an energy range 2- or 3-times wider than the energy range of the EGS4 simulation.

- TEST means "Plot the fitted functions" in the line printer image. (Not changed usually.)

- PWLF means "select a piecewise linear fit". (Not changed usually.)

- DECK means "punch fit data and other useful parameters". (Not usually changed.)

Next, input data for a single-atom gas (Not a molecule gas) is described. As an example, the first 4 lines of the input for Xe-gas (1 atm, 0 °C) is shown. [3] (The input after ENER is not shown, since it is the same as in the previous example.)

```
ELEM
 &INP RHO=5.89E-3,GASP=1.,IAPRIM=1 &END
XENON-GAS                       XE-GAS
XE
```

- 'RHO' specifies the density (g/cm^3) of the compound (at STP for gases). STP: Standard Temperature and Pressure= 0 °C, 1 atm.

- 'GASP' Optional. Defines the state of the compound: zero (default) for a solid or liquid;, otherwise, the value gives the gas pressure (atm). By "GASP>0", PEGS4 recognizes the material is the gas. GASP is in atm at the standard temperature (0 °C). If the gas temperature is different from the standard temperature, calculate and input the pressure of that gas when the gas temperature is changed as the standard temperature while the gas volume is not changed.

[3] While GASP is not shown in Table A3.3.1 of SLAC265, GASP is accepted for ELEM like in COMP and in MIXT.

- 'XENON-GAS' is an identifier assigned to a data set to be produced. EGS4 uses this identifier when each material is assigned to each region.
- 'XE-GAS' is an identifier for Xe gas in Table 2.13.2 of SLAC-265.
- 'XE' in the 4th line is an atomic symbol to specify an element.

2 Input file for compound

As an example of input for compound material data, the input for acryl (PMMA: Polymethyl methacrylate) is shown:

```
COMP
 &INP NE=3,RHO=1.19,PZ=5.,8.,2.,IAPRIM=1 &END
PMMA-IAPRIM                    PMMA
C  H  O
ENER
 &INP AE=0.521,UE=50.511,AP=0.01,UP=50.0 &END
TEST
 &INP &END
PWLF
 &INP &END
DECK
 &INP &END
```

- 'COMP' in the 1st line means "select a material that is a compound".
- 'NE=3' means the number of elements in the compound is 3.
- 'RHO' is the density (g/cm^3) of the compound (at STP for gases).
- 'PZ' indicates the relative numbers of atoms in a compound.
- The 3rd line is written in the same way as ELEM. If the compound is contained in Table 2.13.2 of SLAC-265, input the compound name from the 31st column. This input is ignored if the compound identifier does not appear in Table 2.13.2, as in this example.
- In the 4th line, input the atomic symbols in the same order as PZ in the (A2,1X) format.
- The 5th line and below that line is the same as ELEM.

As an example of a gaseous compound, the input for CO_2 gas is shown. Only the beginning 4 lines are shown, since the 5th line and below the 5th line are the same as in the previous example. For 0 °C and 1 atm,

```
COMP
 &INP NE=2,RHO=1.977E-3,GASP=1.,PZ=1.,2.,IAPRIM=1 &END
CO2                            CO2-GAS
C  O
```

For 0 °C and 10 atm,

```
COMP
 &INP NE=2,RHO=1.977E-3,GASP=10.,PZ=1.,2.,IAPRIM=1 &END
CO2-10atm                      CO2-GAS
C  O
```

For 20 °C and 1 atm,

```
COMP
 &INP NE=2,RHO=1.977E-3,GASP=0.93174,PZ=1.,2.,IAPRIM=1 &END
CO2-20degree                  CO2-GAS
C  O
```

- RHO is the density at STP (always).
- In the last example, 0.93174 atm （=273 °C/293 °C） is used as GASP. This pressure is obtained when a gas of 20 °C and 1 atm is cooled down to 0 °C while the volume remains unchanged.
- 'CO2-10atm' and 'CO2-20degree' are identifiers which a user can freely determine.
- 'CO2-GAS' is an identifier of CO_2 gas in Table 2.13.2 of SLAC-265.

Next, for hydrogen gas (0 °C、1 atm),

```
COMP
 &COMP NE=2,RHO=8.99E-5,GASP=1.,IAPRIM=1,PZ=1.,1. &END
H2-GAS                        H2-GAS
H  H
```

- In PEGS4, a molecular gas comprising a single element (ex. H_2) is treated as a compound. [4]
- 'NE=2' must be input. Because 'NE=1' causes an error.
- 'H2-GAS' is an identifier of hydrogen gas in the density-effect table (SLAC-265 Table 2.13.2).

3 Input for a mixture

The beginning 4 lines of the input for lead glass is shown as an example of mixture material data. (See input for an element for the 5th line and below that line.)

```
MIXT
 &INP NE=5,RHO=3.61,RHOZ=41.8,21.0,29.0,5.0,2.2,IAPRIM=1 &END
LEAD GLASS
PB SI O  K  NA
```

- 'MIXT' means "select a material that is a mixture."
- 'NE=5' indicates that the number of elements in a mixture is 5.
- 'RHO' is the density (g/cm^3) of the mixture (at STP for gases).
- 'RHOZ' is the relative amount of atoms in the mixture (by weight).
- 4th line. Atomic symbol in the same order as RHOZ in the (A2,1X) format.

The beginning 5 lines of input for air (20 °C、1 atm) are shown as an example of a gaseous mixture.

[4] When the SSB coefficient is not given for a single-element molecular gas, Sternheimer-Peierls' general formula is used. In the Sternheimer-Peierls' general formula, a molecular gas is treated as a compound. In that case the material must be treated as a compound.

On the other hand, when the SSB coefficient is used, as in this example, a single-element molecular gas can be treated either as an element or a compound.

```
MIXT
 &INP NE=3,RHO=1.2929E-3,GASP=0.93174,RHOZ=0.75575,0.23143,0.01282,
 IAPRIM=1 &END
AIR                              AIR-GAS
N  O  AR
```

- 'RHO' is the density (g/cm^3) at STP.
- 'AIR-GAS' is an identifier of air in Table 2.13.2 in SLAC-265.
- Ar is an important component in energy region below 100 keV.

4 CALL Option

CALL option is an input option to output a specific evaluated value in material data. For example, the next input specifies the output mean free path for a 49.99-MeV photon.

```
ELEM
 &INP IAPRIM=1 &END
PB
PB
CALL
 &INP XP(1)=49.99 &END
GMFP
```

The output for this input is,

```
OPT=CALL
FUNCTIONCALL:    1.95522  = GMFP  OF 49.9900
```

Be careful concerning the unit of GMFP. It is not cm, but radiation length (r.l.).

5 How to use a PEGS4 improvement developed in Japan

- Cross section based on PHOTX data [5].
  - Use pgs4phtx.dat in egs4unix_kek. Comment out

    `OPEN(8,file='pegs4pepr.dat');`

    in pegs4.mortran. Then un-comment out

    `OPEN(8,file='pegs4phtx.dat');`
  - It is better to use pegs4phtx.dat in general. pegs4pepr.dat is used as a default just to maintain the condition of the original distribution of the EGS4 code.
- Sternheimer density effect coefficient by Sternheimer, Berger and Seltzer in 1984. [6]
  - Use pegs4.mortran in,

    `ftp.kek.jp:/kek/kek_egs4/pegs4/pegs4.tar.Z`
  - Specify the material identifier in Table 1 from the 31st column of the 3rd line of the PEGS4 input. (Be careful that the identifier in Table 1 is effective only for this expanded PEGS4! Only the identifier shown in Table 2.13.2 in SLAC-265 is effective for the standard PEGS4.)

[5] Y. Sakamoto, "Photon cross section data PHOTX for PEGS4", Proceedings of the Third EGS4 User's meeting in Japan, (In Japanese) KEK Proceedings, 93-15 (Dec 1993).

[6] H. Hirayama, "Revision of the Sternheimer Density Effect Coefficients in PEGS4", KEK Internal 95-17 (1995).

- Total Compton-scattering cross section for a bound electron, incoherent scattering function and Compton profile. [7]
  - Use pegs4nb.mortran in,

    `ftp.kek.jp:/kek/kek_egs4/lscat/lscat_1.1.tar.Z`
  - Important for photon transport below 100 keV.

[7] Y. Namito et al "LSCAT: Low-Energy Photon-Scattering Expansion for the EGS4 Code", KEK Internal 95-10 (1995).

Table 1 Identifier of medium name for Sternheimer-Seltzer-Berger Coefficients in PEGS4

LABEL	LABEL	LABEL
H2-GAS	RH	PU
H2-LIQUID	PD	AM
HE-GAS	AG	CM
LI	CD	BK
BE	IN	A-150 TE PLASTIC
B	SN	ACETONE
C-2.265 G/CM**3	SB	ACETYLENE
C-2.00 G/CM**3	TE	ADENINE
C-1.70 G/CM**3	I	ADIPOSE TISSUE
N2-GAS	XE-GAS	AIR-GAS
O2-GAS	CS	ALANINE
F	BA	ALUMINIUM OXIDE
NE-GAS	LA	AMBER
NA	CE	AMMONIA
MG	PR	ANILINE
AL	ND	ANTHRACENE
SI	PM	B-100 BONE-EQ. PLASTIC
P	SM	BAKELITE
S	EU	BARIUM FLUORIDE
CL	GD	BARIUM SULFATE
AR-GAS	TB	BENZENE
K	DY	BERYLLIUM OXIDE
CA	HO	BISMUTH GERMANIUM OXIDE
SC	ER	BLOOD (ICRP)
TI	TM	BONE, COMPACT (ICRU)
V	YB	BONE CORTICAL (ICRP)
CR	LU	BORON CARBIDE
MN	HF	BORN OXIDE
FE	TA	BRAIN (ICRP)
CO	W	BUTANE
NI	RE	N-BUTYL ALCHOL
CU	OS	C-552 AIR-EQ. PLASTIC
ZN	IR	CADMIUM TELLURIDE
GA	PT	CADMIUM TUNGSTATE
GE	AU	CALCIUM CARBONITE
AS	HG	CALCIUM FLUORIDE
SE	TL	CALCIUM OXIDE
BR	PB	CALCIUM SULFATE
KR-GAS	BI	CALCIUM TUNGSTATE
RB	PO	CARBON DIOXIDE
SR	RN-GAS	CARBON TETRACHLORIDE
Y	RA	CELLOPHANE
ZR	AC	CELLULOSE ACETATE BUTYRA
NI	TH	CELLULOSE NITRATE
MO	PA	CERIC SURFARE DOSIMETER
TC	U	CESIUM FLUORIDE
RU	NP	CESIUM IODIME

Table 1 Identifier of medium name for Sternheimer-Seltzer-Berger Coefficients in PEGS4

LABEL	LABEL	LABEL
CHLOROBENZENE	LITHIUM TETRABORATE	POTASSIUM IODINE
CHLOROFORM	LUNG (ICRP)	POTASSIUM OXIDE
CONCRETE, PORTLAND	M3 WAX	PROPANE
CYCLOHEXANE	MAGNESIUM CARBONATE	PROPANE, LIQUID
1,2-DICHLOROBENZENE	MANESIUM FLUORIDE	N-PROPYL ALCOHOL
DICHLORODIETHYL ETHER	MAGNESIUM OXIDE	PYRIDINE
1,2-DICHLOROETHANE	MAGNESIUM TETRABORATE	RUBBER, BUTYL
DIETHYL ETHER	MERCURIC IODIDE	RUBBER, NATURAL
N,N-DIMETHYL FORMAMIDE	METHANE	RUBBER, NEOPRENE
DIMETHYL SULFOXIDE	METHANOL	SILICON DIOXIDE
ETHANE	MIX D WAX	SILVER BROMIDE
ETHYL ALCOHOL	MS20 TISSUE SUBSTITUTE	SILVER CHLORIDE
ETHYL CELLULOSE	MUSCLE, SKELETAL (ICRP)	SILVER HALIDES IN EMUL.
ETHYLENE	MUSCLE, STRIATED (ICRU)	SILVER IODIDE
EYE LENS (ICRP)	MUSCLE-EQ. LIQ. W SUCROS	SKIN (ICRP)
FERRIC OXIDE	MUSCLE-EQ. LIQ. W/O SUCR	SODIUM CARBONATE
FERROBORIDE	NAPTHALENE	SODIUM IODIDE
FERROUS OXIDE	NITROBENZENE	SODIUM MONOXIDE
FERROUS SULFATE DOSIMETE	NITROUS OXIDE	SODIUM NITRATE
FREON-12	NYLON, DU PONT	STILBENE
FREON-12B2	NYLON, TYPE 6 AND 6/6	SUCROSE
FREON-13	NYLON, TYPE 6/10	TRRPHENYL
FREON-13B1	NYLON, TYPE 11	TESTES (ICRP)
FREON-13I1	OCTANE, LIQUID	TETRACHLOROETHTLENE
GADOLINIUM OXYSULFIDE	PARAFFIN WAX	THALLIUM CHLORIDE
GALLIUM ARSENIDE	N-PENTANE	TISSUE, SOFT (ICRP)
GEL IN PHOTOGRAPHIC EMUL	PHOTOGRAPHIC EMULSION	ICRU FOUR-COMP. TISSUE
GLASS (PYREX)	PLASTIC SCINT.	TISSUE-EQ. GAS (METHANE)
GLASS, LEAD	PLUTONIUM DIOXIDE	TISSUE-EQ. GAS (PROPANE)
GLASS, PLATE	POLYCRYLONITRILE	TITANIUM DIOXIDE
GLUCOSE	POLYCARBONATE	TOLUEN
GLUTAMINE	POLYCHLOROSTYRWNE	TRICHLOROETHYLENE
GLYCEROL	POLYETHYLENE	TRIETHYL PHOSPHATE
GUANINE	MYLAR	TUNGSTEN HEXAFLUORIDE
GYPSUM, PLASTER OF PARIS	LUCITE	URANIUM DICARBIDE
N-HEPTANE	POLYOXYMETHYLENE	URANIUM MONOCARBIDE
N-HEXANE	POLYPROPYLENE	URANIUM OXIDE
KAPTON, POLYIMIDE FILM	POLYSTYRENE	UREA
LANTHANUM OXYBROMIDE	TEFLON	VALINE
LANTHANUM OXYSULFIDE	POLYTRIFLUOROCHLOROETHY.	VITON
LEAD OXIDE	POLYVINYL ACETATE	WATER, LIQUID
LITHIUM AMIDE	POLYVINYL ALCOHOL	WATER VAPOR
LITHIUM CARBONATE	POLYVINYL BUTYRAL	XYLENE
LITHIUM FLUORIDE	POLYVINYL CHLORIDE	
LITHIUM HYDRIDE	SARAN	
LITHIUM IODIDE	PLOYVINYLIDENE FLUORIDE	
LITHIUM OXIDE	POLYVINYL PYRROLIDONE	

How to Write User Code

1 Introduction

Although it is possible to write the User Code following the explanations given in SLAC-265, this way is generally not easy, especially for beginners. The most effective way is to modify the User Code written for a similar problem.

In this note, basic constructions of the User Code are explained using "ucnai3p.mor", which is used to calculate the response of a NaI detector of cylinder-slab geometry(Fig. 1).

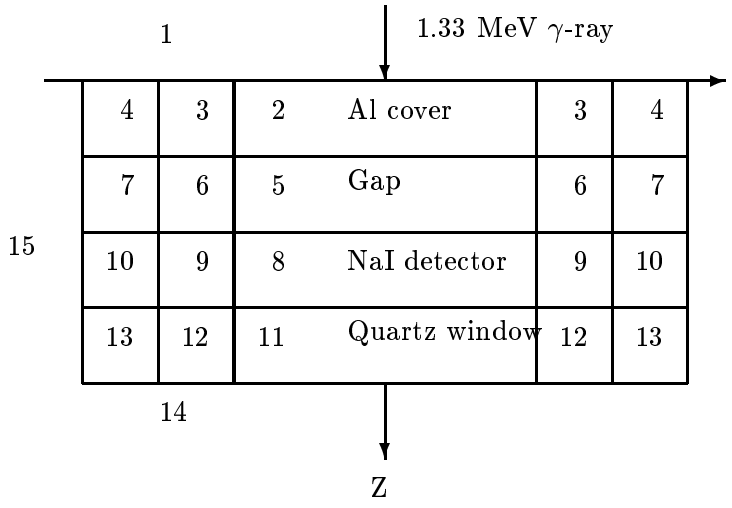


Fig. 1 Geomtry of ucnai3p.mor.

2,3,4,7,10,13 – Al

11,12 – Quartz

5,6,9 – Gap

8 – NaI

2 STEP 1

If a user wishes to modify the macros used in EGS4, put over-ride macros here. Mortran uses the most recent pattern if the same macro pattern is defined differently. The User Code is positioned after egs4mac.mortran (or egs4mac.mor for a PC). Therefore, if over-ride macros are defined in the User Code, they become effective as the macro definitions. This means that a user can modify EGS4, itself, by using these over-ride macros.

The first macro calculates the shortest distance to the boundary, which is necessary in PRESTA. This macro is different for the geometry used and, therefore, must be modified depending on the geometry.

```
REPLACE{$CALL-HOWNEAR(#);} WITH
  {$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY({P1});}
 "                    ****                                    "
 "THIS IS THE MACRO THAT SHOULD RETURN THE CLOSEST PERPENDICULAR    "
 "DISTANCE TO ANY SURFACE WHICH FORMS A BOUNDARY FOR THE CURRENT    "
 "REGION. IN THIS APPLICATION IT IS REPLACED BY THE MACRO FOLLOWING "
 "WHICH IS SPECIALIZED FOR THE SLAC       "
 "                                                  ****       "
 "CYLINDRICAL-PLANE GEOMETRY.                                   "
 "IT IS THE USER'S RESPONSIBILITY TO PROVIDE THIS MACRO FOR HIS OWN "
 "GEOMETRY.   "
 "++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++   "

 ; "BUFFER FLUSH"
 REPLACE{$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY(#);} WITH
 "       ==================**************************"
    {;ZL=Z(NP);RL=SQRT(X(NP)**2+Y(NP)**2);
    $GET-JR-JZ(IRL);
    ZLEFT=ZL-PCOORD(3,JZ);ZRIGHT=PCOORD(3,JZ+1)-ZL;ROUT=CYRAD(JR)-RL;
    {P1}=MIN(ZLEFT,ZRIGHT,ROUT);
    IF(JR.NE.1) [RIN=RL-CYRAD(JR-1);{P1}=MIN({P1},RIN);]}
```

```
"THIS ROUTINE IS INTENDED TO BE USED TO CALCULATE THE MINIMUM      "
"PERPENDICLAR TO THE NEAREST BOUNDING SURFACE. THIS VERSION IS     "
"SPECIALLY DESIGNED FOR THE SLAC CYLINDRICAL GEOMETRY PACKAGE.     "
"A DIFFERENT VERSION IS NEEDED FOR OTHER    "
"****                                                              "
"GEOMETRY PACKAGES.                                                "

; "BUFFER FLUSH"
"SLAC MACRO TO GET CYLINDER (JR) AND SLAB (JZ) ZONES FROM REGION   "
"NUMBER"
"*****                                                             "
REPLACE {$GET-JR-JZ(#);} WITH
"          =============="
  {;JZ=({P1}-2)/NCYLP+1;JR={P1}-1-NCYLP*(JZ-1);}
; "BUFFER FLUSH"
"GEOMETRICAL INFORMATION"
"SLAC DEFINITION OF /GEOM/ AND RE-DEFINITIONS FOR /CYLDTA/ AND     "
"/PLADTA/        "
```

The following over-ride macros are defined so as to use variables which are necessary for PRESTA in SUBROUTINE ELECTR.

```
"****                                                              "
REPLACE {;COMIN/GEOM/;} WITH {;COMIN/CYLDTA,PLADTA/;}
"       =============="

REPLACE {;COMIN/CYLDTA/;} WITH
"       ================"
  {;COMMON/CYLDTA/CYRAD($MXCYLS),CYRAD2($MXCYLS),NCYLP;}
"NOTE: CYRAD MUST BE PROVIDED IN ADDITION TO CYRAD2 (USUALLY IN    "
"MAIN)"

REPLACE {;COMIN/PLADTA/;} WITH
"       ================"
   {;COMMON/PLADTA/PCOORD(3,$MXPLNS),PNORM(3,$MXPLNS);}
```

It is better to also put macros defined by the user in this step. In ucnai3p.mor, COMMON/TOTALS (related to score variables), COMMON/PASSIT (related to geometry) and COMMON/LINES (related to print out intermediate results) are defined as follows:

```
"COMMON to define variables to score at AUSGAB"
"DEPE:deposited energy inside the detector"
"DELTAE:energy bin width in MeV"
"SPG:Gamma spectrum, SPE:Electron spectrum, SPP:Positron spectrum"

REPLACE {;COMIN/TOTALS/;} WITH
{;COMMON/TOTALS/DEPE,DELTAE,SPG($NDET,$NEBIN),SPE($NDET,$NEBIN),
SPP($NDET,$NEBIN);}

"COMMON of print-out parameter"

REPLACE {;COMIN/LINES/;} WITH
{;COMMON/LINES/NLINES,NWRITE,NCOUNT,ILINES;}

"COMMON of geometry related parameter"

REPLACE {;COMIN/PASSIT/;} WITH
   {;COMMON/PASSIT/NREG,NPLAN;}
```

$PARAMETER statements define the values of the variables used in the program. If the arguments of the variables are defined by $PARAMETER statements, their modification can be done only at PARAMETER statements.

The following $PARAMETER statements are used for defining the arguments of variables:

```
PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
PARAMETER $NBATCH=50;  "Number of batch"
PARAMETER $NEBIN=50;   "Number of energy bin"
PARAMETER $NDET=1;     "Number of detector"
```

Declarations related to COMMON and DIMENSION are used in the main program after PARAMETER statements. Since COMMONs in EGS4 are defined in the form of macros, the COMMON parts are expressed as:

```
;COMIN/DEBUG,BOUNDS,BREMPR,EDGE,ELECIN,ETALY1,PLADTA,LINES,MEDIA,
MISC,NTALY1,PASSIT,RANDOM,STACK,THRESH,TOTALS,UPHIOT,USEFUL,USER/;
```

LINES, PASSIT, TOTALS above are the ones defined in this User Code as mentioned previously.

The next step is defining the material used in the User Code. A material name is defined by 24 characters. The first argument of a MADARR must be 24. The second argument is the number of materials used in the User Code. In this User Code, this number is 3.

Although any file names can be assigned for UNIT 6, 8, mortjob.xsec (or mortjob.xse in the case of PC) must be assigned as the file name of UNIT 12 when the user runs the EGS4 program using egs4run.

3 STEP 2

The variables used in SUBROUTINE HATCH must be defined together with the number of planes, cylinders and regions shown below. NCYLP is a variable defined for use in PRESTA. NMED is the number of materials used, and must be the same as the second argument of MEDARR.

```
NMED=3; "NUMBER OF MEDIA"

DO J=1,NMED [
DO I=1,24 [MEDIA(I,J)=MEDARR(I,J);]]

NPLAN=$MXPLNS;    "NUMBER OF PLANES"
NCYL=$MXCYLS;    "NUMBER OF CYLINDER"
NCYLP=NCYL;   "*PRESTA*"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
"                       REGION)                                    "
IRZ=NREG-3;
"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0;   "VACUUM REGIONS"

/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
"                                  Al region"
MED(8)=1;   "NaI(Tl) detector region"
/MED(11),MED(12)/=3;  "Quartz region"
/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0;  "Vacuum region inside case"

IEDGFL(8)=53;  "53:Atomic number of I"
"               0:K-X ray of I is not produced"
```

The numbers of regions, material of each region and ECUT, PCUT of each region are defined here. An atomic number to produce K X-rays is also defined here to each region if it is desired.

AE and AP are used in PEGS4 as the cut-off energy of the material. Electron or positron scattering producing a secondary electron less than AE and bremsstrahlung producing a photon less than AP are included in a continuous slowing down process. If the energy of a particle becomes smaller than ECUT(electron/positron) or PCUT(photon), all kinetic energy of that particle is treated as energy deposition at the point. In the case that ECUT and PCUT is not assigned to each region, AE and AP are assigned as ECUT and PCUT, respectively.

In EGS4, a compound or mixture material is also treated like an element material. It is not easy to treat K X-rays of a compound or mixture generally due to this treatment. One approximate way is to assign the atomic number of the element which is most effective for K X-ray production. In this User Code, the atomic number of I is used as that of NaI (IEDGFL(4)=53;).

3.1 Set Up Options

Various improvements have been made after the release of EGS4. Most of them are in the form of macros, and can be applied by setting up the corresponding flag, like `IEDGFL`.

The following are introductions for some important flags:

1. Emitting angle of bremsstrahlung[1]
   In the default EGS4, bremsstrahlung is emitted in the direction of the critical angle, $\theta(= m_0/E_0$ radian; m_0, rest mass energy of an electron/positron; E_0,total energy of an electron/positron). If `IBRDST=1` is set, the emitted angle is sampled within the critical angle.

```
"THE FOLLOWING REPLACES THE EGS4 DEFAULT $SET-BREMS-ANGLE MACRO   "
"IT'S USE REQUIRES AN ASSOCIATE MACRO $SET-BREM-REJECTION-FUNCTION"
"DEFINED BELOW                                                    "
"                                                                 "
"USAGE: IBRDST=0 => EGS4 DEFAULT ANGLE SELECTION                  "
"       IBRDST=1 => KOCH AND MOTZ (1959) EQ. 2BS ANGLE SELECTION  "
```

   `COMIN/BREMPRR/;` must be included at main to use this flag.

2. Emitting angle of an electron and a positron after pair-production[2]
   A similar treatment is applied to pair production in the default EGS4. `IPRDST` is used to sample the emitting angle more precisely, as shown below:

```
"USAGE: IPRDST=0 => EGS4 DEFAULT ANGLE SELECTION                  "
"       IPRDST=1 => LOWEST ORDER ANGULAR DISTRIBUTION             "
"                                                                 "
"          d(Probability)          sin(theta)                     "
"         --------------- = ------------------------------------"
"            d(theta)        2*P[E_total - P*cos(theta)]**2       "
"                                                                 "
"       IPRDST=2 => MOTZ, OLSEN AND KOCH (1969) EQ. 3D-2003       "
"                   IF IPRDST IS NON-ZERO AND E_PHOTON < $BHPAIR  "
"                   THEN IPRDST=1 DISTRIBUTION IS USED            "
```

   `IPRDST` is also included in `COMIN/BREMPRR/;`.

3. Angular distribution of photoelectrons[3]
   A photoelectron is emitted in the direction of the incident photon in the default EGS4. `IPHTER` must be asigned for each region to apply the angular distribution for photoelectrons.

```
"       IPHTER       REGION DEPENDENT ARRAY FOR SWITCHING ON     "
"                    PHOTOELECTRON ANGULAR DISTRIBUTION          "
"                    DEFAULT(0)-NO SAMPLING, (1)-SAMPLING        "
```

   `COMIN/USER/;` must be included to use `IPHTER` in the main program.

4. Bremsstrahlung Splitting[1]
   Bremsstrahlung splitting is a useful variance reduction technique to calculate the contribution of bremsstrahlung in the situation that the bremsstrahlung production probability is small, like in the case of low energy electrons. `IBRSPL=1` and `NBRSPL` must be set to use the bremsstrahlung splitting option.

```
"THIS MACRO PLACES ADDITIONAL BREMSSTRAHLUNG PHOTONS ON THE STACK   "
"RESETTING PARTICLE WEIGHTS TO MAKE THE GAME FAIR. THREE USER INPUTS"
"ARE REQUIRED:                                                     "
"                                                                  "
"IBRSPL = 0 => NO ADDITIONAL BREMSSTRAHLUNG PHOTONS (DEFAULT)      "
```

```
"       = 1 => PERFORM BREMSSTRAHLUNG SPLITTING                          "
"NBRSPL = NUMBER OF BREMSSTRAHLUNG PHOTONS CREATED/INTERACTION           "
"FBRSPL = 1/NBRSPL (USED TO ADJUST THE PARTICLE WEIGHTS)                 "
"                                                                        "
"NBRSPL AND FBRSPL ARE CHANGED DYNAMICALLY IF STACK OVERFLOW MIGHT       "
"OCCUR                                                                   "
"                                                                        "
"THIS MACRO IS INVOKED AFTER THE FIRST CALL THE SUBROUTINE BREMS         "
```

`COMIN/BREMPRR/;` must be included to use this option.

4 STEP 3

At step 3, `SUBROUTINE HATCH` is called and material data used in the User Code are read from the material data file. It is better to print out information concerning the materials used (name, density, radiation length, cut-off energy). The name and cut-off energy of each region are also useful to check the program. If the region number is large, it is better to print out only that information necessary to check the material assignment.

If `SUBROUTINE PHOTO` is the modified version which includes `K X-ray` productions, the following statement must be included to call `SUBROUTINE EDGSET` depending the values of `IEDGFL`:

```
DO I=1,NREG [IF(IEDGFL(I).NE.0) [CALL EDGSET(NREG); EXIT;]]
```

5 STEP 4

Various data related to the geometrical expression are defined at this step. The coordinates (`PCOORD`) and normals(`PNORM`) of the plane are defined for planes. The radius (`CYRAD`) and radius squared (`CYRAD2`) are defined for cylinders. It is better to print out the defined information for checking the geometry.

6 STEP 5

The initialize variables used in `AUSGAB` are as follow:

```
NCOUNT=0;  "PARTICLE HISTORY COUNTER"
ILINES=0;  "INITIALIZE LINE-OUTPUT COUNTER"
DEPE=0.D0; "ZERO THE ENERGY DEPOSITION AT SCINTILLATOR"
/PEF,TEF/=0.0;  "Zero the efficiency"

DO J=1,$NEBIN [PH(J)=0.0;]  "Zero the pulse-height"
DO ND=1,$NDET [
DO J=1,$NEBIN [
/SPG(ND,J),SPE(ND,J),SPP(ND,J)/=0.0; "Zero the spectrum"
]]
```

`NCOUNT` and `ILINES` are variables used to control the output of intermediate results.

7 STEP 6

Define the parameters of the incident particles as follows:

```
CALL ECNSV1(0,NREG,TOTKE);"  INITIALIZE ESUM ARRAY FOR ENERGY"
"                            CONSERVATION CALCULATION."
"                            NREG=NUMBER OF REGIONS"
"                            TOTKE=TOTAL KE (DUMMY VARIABLE HERE)"
"                                  (MUST BE REAL*8)"

CALL NTALLY(0,NREG);
```

```
IQI=0;  "INCIDENT PARTICLE"

EI=1.33 +ABS(IQI)*PRM;   "TOTAL ENERGY OF PARTICLE (MEV) "

AVAILE=EI + IQI*PRM;  "AVAILABLE K.E. (MEV) (MUST BE REAL*8)"
EKIN=AVAILE;
ECUTMN=ECUT(4); EKO=EKIN;  "*PRESTA*"
$PRESTA-INPUTS;  "INPUT THE *PRESTA* VARIABLES"

DELTAE=0.05;  "Energy bin of response"

XI=0.0; YI=0.0; ZI=0.0;  "STARTING COORDINATES (CM)"
UI=0.0; VI=0.0; WI=1.0;  "INCIDENT DIRECTION COSINES"
IRI=2;  "ENTRANCE REGION DEFINITION"
WTI=1.0;  "WEIGHT FACTOR OF UNITY"

IDINC=-1; "AN IDENTIFIER (LIKE IARG) TO MARK INCIDENT PARTICLES"

IXXST=17847465;

IXX=IXXST;  "INITIALIZED RANDOM NUMBER WITH STARTING SEED"

NWRITE=10;  "NUMBER OF INCIDENT CASES TO PRINT OUT"

NCASES=$NCASES; "MAXIMUM NUMBER OF INCIDENT CASES TO RUN"
NBATCH=$NBATCH;  "NUMBER OF BATCH"
NCASPB=NCASES/NBATCH; "NUMBER OF CASES PER BATCH"
NOFBAT=0;  "NUMBER OF BATCH FINISHED"

NLINES=15; "NUMBER OF LINES TO PRINT OUT"
```

IQI:type of particle, EI:total energy, XI,YI,ZI:incident position,
UI,VI,WI:direction cosine, IRI:incident region number,
WTI:weight of particle (in ordinary case =1)

`ECUTMN=ECUT(4); EKO=EKIN;` is defined to initialize `PRESTA` by the macro `$PRESTA-INPUTS`. In addition, the following variables are also defined in this step:

`IDINC=-1`:variable to indicate the incident particle,
`IXXST`:initial seed of random number,
output condition of intermediate results,
`NCASES`:history number,
`NBATCH`:batch number if the calculation is divided into batch,
`NCASPB`:history number per batch

8 STEP 7

This step is the main part of the User Code calling `SUBROUTINE SHOWER` `NCASES`-times. If incident particles some energy, direction or space distribution, the routine to sample these parameters must be put before `CALL SHOWER;`. `IRI`, which is the region of the source particle, must be modified if it depends on the sampling results.

8.1 Statistical analysis

Assume that x is a quantity we calculate during the course of a Monte Carlo calculation, *i.e.* a scoring variable. The output of a Monte Carlo calculation is usually useless unless we can ascribe a probability error to it. There are 2 conventional approaches to calculating the probable error.

8.1.1 Method used in MCNP

- Assume that the calculation calls for N "incident" particle histories.
- Assume that x_i is the result at the i-th history.
- Calculate the mean value of x :

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{1}$$

- Estimate the variance associated with the distribution of x_i:

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \simeq \overline{x^2} - (\overline{x})^2 \quad (\overline{x^2} = \frac{1}{N}\sum_{i=1}^{N} x_i^2). \tag{2}$$

- Estimate the variance associated with the distribution of $\overline{x}$:

$$s_{\overline{x}}^2 = \frac{1}{N}s^2 \simeq \frac{1}{N}[\overline{x^2} - (\overline{x})^2] \tag{3}$$

- Report the statistical error as:

$$R = s_{\overline{x}}/\overline{x} \simeq [\frac{1}{N}(\frac{\overline{x^2}}{\overline{x}^2} - 1)]^{1/2} \tag{4}$$

8.1.2 Method used in MORSE-CG

- Assume that the calculation calls for N "incident" particle histories.
- Split the "N" histories into n statistical batches of N/n histories each. The calculated quantity for each of these batches is called x_i.
- Calculate the mean value of x:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{n} x_i \tag{5}$$

- Estimate the variance associate with the distribution of the x_i:

$$s_x^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i^2 - \overline{x}^2) \tag{6}$$

- The estimated variance of $\overline{x}$ is the standard variance of the mean:

$$s_{\overline{x}}^2 = \frac{s_x^2}{n} \tag{7}$$

- Report FSD(fractional standard deviation) as the statistical error:

$$\text{FSD} = s_{\overline{x}}/\overline{x} \tag{8}$$

8.1.3 Treatments in ucnai3p.mor for a statistical analysis

ucnai3p.mor calculates FSD based on the method used in MORCE-CG.

The histories(NCASES) are divided into NBATCH statistical batches of NCASPB(=NCASES/NBATCH) histories each. The response, peak and total efficiency obtained from the energy deposition inside the detector are calculated at each history.

```
"If some energy is deposited inside detector add pulse-height"
"and efficiency"

IF(DEPE.GT.0.D0) [
IE=DEPE/DELTAE+1;
IF(IE.LE.$NEBIN) [PH(IE)=PH(IE)+1.0;]
IF(DEPE.GE.EI*0.999) [PEF=PEF+1.0;]
TEF=TEF+1;]
DEPE=0.D0;
```

The mean value $\overline{x}$ is calculated after the end of each batch.

```
"Calculate average value for this BATCH"
DO IE=1,$NEBIN [
PHPB(IE,NOFBAT)=PH(IE)/NCASPB;
PH(IE)=0.0;
]
PEFPB(NOFBAT)=PEF/NCASPB;
TEFPB(NOFBAT)=TEF/NCASPB;
/PEF,TEF/=0.0;

DO ND=1,$NDET [
DO IE=1,$NEBIN [
SPGPB(ND,IE,NOFBAT)=SPG(ND,IE)/NCASPB; "Gamma spectrum for this BATCH"
SPEPB(ND,IE,NOFBAT)=SPE(ND,IE)/NCASPB; "Electron spectrum for this BATCH"
SPPPB(ND,IE,NOFBAT)=SPP(ND,IE)/NCASPB; "Positron spectrum for this BATCH"
/SPG(ND,IE),SPE(ND,IE),SPP(ND,IE)/=0.0;
]]
```

9 STEP 8

Analyze the obtained results and output them in this step. It is also better to print out the information about the geometry and source particle.

Calculate the average values and their deviations (x_i) from the results per batch.

In this User Code, the detector response and particle spectrum are presented per source particle per MeV. Therefore, the results are divided by the energy-bin width(DELTAE).

```
"Calculate average and its deviation"

/AVPE,DESCI2/=0.0;
DO J=1,NBATCH [
AVPE=AVPE+PEFPB(J)/NBATCH;
DESCI2=DESCI2+PEFPB(J)*PEFPB(J)/NBATCH;
]
SIGPE=SQRT((DESCI2-AVPE*AVPE)/(NBATCH-1));
AVPE=AVPE*100.0;
SIGPE=SIGPE*100.0;
OUTPUT AVPE,SIGPE;(' Peak efficiency =',G15.5,'+-',G15.5,' %');

/AVTE,DESCI2/=0.0;
DO J=1,NBATCH [
AVTE=AVTE+TEFPB(J)/NBATCH;
DESCI2=DESCI2+TEFPB(J)*TEFPB(J)/NBATCH;
]
SIGTE=SQRT((DESCI2-AVTE*AVTE)/(NBATCH-1));
AVTE=AVTE*100.0;
SIGTE=SIGTE*100.0;
OUTPUT AVTE,SIGTE;(' Total efficiency =',G15.5,'+-',G15.5,' %');
```

```
OUTPUT ;(/' Pulse height distribution ');
DO IE=1,$NEBIN [
ELOW=DELTAE*(IE-1);
EUP=DELTAE*IE;
IF(ELOW.GT.EKIN) [EXIT;]

/AVPH,DESCI2/=0.0;
DO J=1,NBATCH [
AVPH=AVPH+PHPB(IE,J)/NBATCH;
DESCI2=DESCI2+PHPB(IE,J)*PHPB(IE,J)/NBATCH;
]
SIGPH=SQRT((DESCI2-AVPH*AVPH)/(NBATCH-1));
AVPH=AVPH/DELTAE;
SIGPH=SIGPH/DELTAE;
OUTPUT EUP,AVPH,SIGPH;
(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
' counts/MeV/incident');
]
```

If User Code used `ECNSV1` or `NTALLY`, statements to output the statistical results obtained by these subroutines.

10 SUBROUTINE AUSGAB

`AUSGAB` is a subroutine used to score the information that user wants to calculate. In ucnai3p.mor, the energy deposition inside a `NaI` detector and the energy spectrum of each particle entering to `NaI` detector from the outside material are scored as follows:

```
IF(MED(IRL).EQ.1) ["particle is inside the detector"
DEPE=DEPE+EDEP*DPWT;  "Add energy deposition"
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
IF(IQ(NP).EQ.0) ["photon"
IE=E(NP)/DELTAE+1;
IF(IE.LE.$NEBIN) [SPG(1,IE)=SPG(1,IE)+DPWT;]]
ELSEIF(IQ(NP).EQ.-1) ["Electron"
IE=(E(NP)-RM)/DELTAE;
IF(IE.LE.$NEBIN) [SPE(1,IE)=SPE(1,IE)+DPWT;]]
ELSE ["Positron"
IE=(E(NP)-RM)/DELTAE;
IF(IE.LE.$NEBIN) [SPP(1,IE)=SPP(1,IE)+DPWT;]]
] "end of entering to detector"
] "end of inside detector"
```

It is possible to replace the first statement by the following one to check if the particle is inside the detector or not:

```
IF(IRL.EQ.8) ["particle is inside the detector"
```

In this method, `IRL.EQ.8` must be changed if the region number is changed, for example, by increasing the plane.

```
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
```

is used to check whether the particle crosses the boundary or not.

For a spectrum not including photons reflected from the quartz window, the above statement must be replaced with:

```
IF((IRL.NE.IROLD.AND.IARG.EQ.0).AND.(W(NP).GT.0.0))
   [ "particle enters into detector"
```

or

```
IF((IRL.NE.IROLD.AND.IARG.EQ.0).AND.(IROLD.EQ.5))
    [ "particle enters into detector"
```

The following statements are used to obtain the information used in ECNSV1 and NTALLY, respectively:

```
"KEEP TRACK OF THE ENERGY DEPOSITION---FOR CONSERVATION PURPOSES"
ESUM(IQ(NP)+2,IRL,IARG+1)=ESUM(IQ(NP)+2,IRL,IARG+1)+EDEP*DPWT;
NSUM(IQ(NP)+2,IRL,IARG+1)=NSUM(IQ(NP)+2,IRL,IARG+1) + 1;
```

The output of intermediate results is done by:

```
IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
OUTPUT E(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP),
  IQ(NP),IRL,IARG;   (7G15.7,3I5);
ILINES=ILINES+1;]
```

11 SUBROUTINE HOWFAR

The function of HOWFAR is to provide information to EGS about the nature of the geometry.

HOWFAR must determine if USTEP will carry the particle past the boundary that the particle is heading towards. If so, then:

1. USTEP must be shrunk to the distance to the boundary, and

2. IRNEW must be set the "new" region in which the particle will end up.

If the particle is in a region designated by the user as a "discard region", the flag DISCARD is set to unity and returns to the calling subprogram.

```
IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]

RETURN;
END;  "END OF SUBROUTINE HOWFAR"
```

How to Code Geometry of EGS4

1 Function of SUBROUTINE HOWFAR

An EGS4 User Code requires:

- SUBROUTINE AUSGAB for scoring results.
- SUBROUTINE HOWFAR to provide information to EGS about the nature of the geometry.

References 1-3 might be useful for knowing how to write HOWFAR. The basic parts of this paper depend on reference 3.

There are three EGS4 variables that play important roles in HOWFAR;: USTEP, IDISC, IRNEW.

These variables are available in COMMON/EPCONT/.

In EGS4, the unit of geometry is called "region". It is possible to assign the same or different material to each region. HOWFAR must be determined if USETP is to carry a particle past the boundary that it is heading towards. If so, then:

1. USTEP must be shrunk to the distance to the boundary
2. IRNEW must be set to the "new" region in which the particle will end up

On the occasion that a particle will end up in a region designated by the user as a "discard region". In such cases the flag DISCARD is set to equal to unity in HOWFAR, and a RETURN is made to the calling subprogram.

Table 1 gives list of the geometry SUBROUTINEs and macros. Although each SUBROUTINE and macro has the same function, the executing time of the macro is faster than that of SUBROUTINE due to its in-line feature.

2 The $PLANE1 Macro

```
$PLANE1(NPLAN,ISIDE,IHIT,TPLN);
   NPLAN: ID number of plane to be checked.
   ISIDE: 1 region is between origin and outer normal.
         -1 region is not between origin and outer normal.
    IHIT: 1 means particle trajectory will hit plane.
          2 means particle trajectory is parallel to plane.
          0 means particle trajectory is away from plane.
    TPLN: distance to plane if IHIT=1.
```

The plane is defined by its normal vector ((PNORM(I,J),I=1,3)) and coordinates at the intersection point of the normal vector and the plane ((PCOORD(I,J),I=1,3)). Both variables are available in COMMON/PLADTA/.

Consider two parallel planes separating three regions, as in Fig.1.

The regions are identified by numbers 22, 23 and 24 and the planes by numbers 6 and 7.

Assuming that planes 6 and 7 are located at z=30.0cm and z=45.0cm, respectively, we have:

```
PCOORD(1,6)=0.0;  PCOORD(2,6)=0.0;  PCOORD(3,6)=30.0;
PNORM(1,6)=0.0;   PNORM(2,6)=0.0;   PNORM(3,6)=1.0;

PCOORD(1,7)=0.0;  PCOORD(2,7)=0.0;  PCOORD(3,7)=45.0;
PNORM(1,7)=0.0;   PNORM(2,7)=0.0;   PNORM(3,7)=1.0;
```

Table 1 Lists of geometry SUBROUTINEs and macros

SUBROUTINE	Function	MACRO
PLANE1	Determines if the particle trajectory strikes a plane surface Returns trajectory distance (TPLN).	$PLANE1
CYLNDR	Determines if the particle trajectory strikes a cylindrical surface. Returns trajectory distance (TCYL).	$CYLNDR
CONE	Determines if the particle trajectory strikes a conical surface. Returns trajectory distance (TCONE).	$CONE
SPHERE	Determines if the particle trajectory strikes a spherical surface. Returns trajectory distance (TSPH).	$SPHRE
CHGTR	Change USTEP and IRNEW whenever USTEP is larger than the trajectory distance (TPLN, TCYL, TCONE, TSPH).	$CHGTR
FINVAL	Determines the coordinates of the particle trajectory at the point of an intersection with a given surface.	$FINVAL
PLAN2P	Determines the intersection point for two parallel planes by calling PLANE1 twice (when necessary) and CHGTR if a plane is hit.	$PLAN2P
PLAN2X	Determines the intersection point for two crossing planes by calling PLANE1 twice (always) and CHGTR if a plane is hit.	$PLAN2X
CYL2	Similar to PLAN2P, but for concentric cylinders.	$CYL2
SPH2	Similar to PLAN2P, but for concentric spheres	$SPH2
CON2	Similar to PLAN2P, but for concentric cone.	$CON2

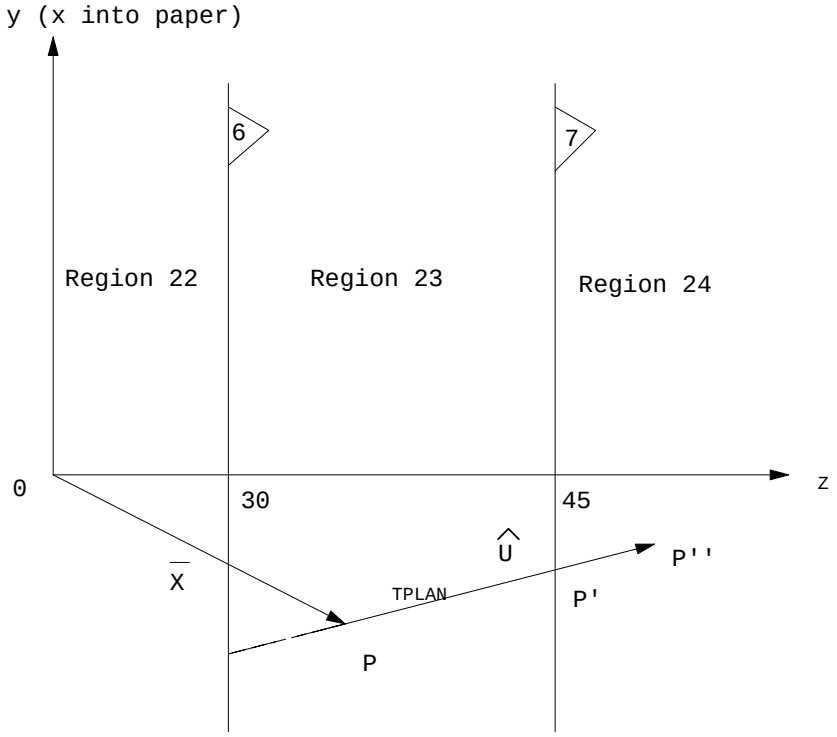


Fig. 1

If particles are initially started in region 23, and discarded when they leave this region, the following HOWFAR will work nicely with EGS4:

```
SUBROUTINE HOWFAR;
COMIN/EPCONT,PLADTA,STACK/;

IRL=IR(NP);
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]
ELSE ["Track particles within region 23"
  $PLANE1(7,1,IHIT,TPLN);  "Check upstream plane first"
  IF(IHIT.EQ.1) ["Surface is hit---make change if necessary"
    $CHGTR(TPLN,24);]
  ELSEIF (IHIT.EQ.0) ["Heading backwards"
    $PLANE1(6,-1,IHIT,TPLN); "To get TPLN-value (IHIT=1, must)"
    $CHGTR(TPLN,22); "Make changes if necessary"
    ]
  ]
RETURN;
END;
```

In the above routine $CHGTR does the following:

- If TPLN.LE.USTEP then USTEP=TPLN and IRNEW=24 (or 22).
- Otherwise, nothing is done.

```
REPLACE{$CHGTR(#,#);} WITH
   {;IF({P1}.LE. USTEP) [USTEP={P1}; IRNEW={P2};]}
```

3 The $PLAN2P Macro

The HOWFAR example that we have been following can be simplified even further with the aide of $PLAN2P:

```
SUBROUTINE HOWFAR;
COMIN/EPCONT,PLADTA,STACK/;

IRL=IR(NP);
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]
ELSE ["Track particles within region 23"
  $PLANE2P(7,24,1,6,22,-1);
  ]
RETURN;
END;
```

The meanings of the parameters in $PLAN2P are as follows:

```
$PLAN2P(NPL1,NRG1,ISIDE1,NPL2,NRG2,ISIDE2);
    NPL1: ID number of first plane to be checked.
    NRG1: region to go into if first plane is intersected by particle.
  ISIDE1: 1 or -1 (same with ISIDE in PLANE1)
    NPL2: ID number of second plane to be checked.
    NRG2: region to go into if second plane is intersected by particle.
  ISIDE2: 1 or -1 (same with ISIDE in PLANE1)
```

The first group of numbers (NPL1,NRG1,ISIDE) ((7,24,1) in the case of the above example) corresponds checking that the downstream plane is equivalent to $PLANE1(7,1,IHIT,TPLN) followed by $CHGTR(TPLN,24).

The second group of numbers (NPL2,NRG2,ISIDE) ((6,22,-1) in the above example) corresponds to checking that the upstream plane is equivalent to $PLANE1(6,-1,IHIT,TPLN) followed by $CHGTR(TPLN,22).

In the case of a pair of crossing planes, $PLAN2X can be used instead of $PLAN2P.

3.1 Multislab Geometry

It is possible to extend the previous HOWFAR for many slabs.

3.1.1 Semi-infinite Plane Geometry

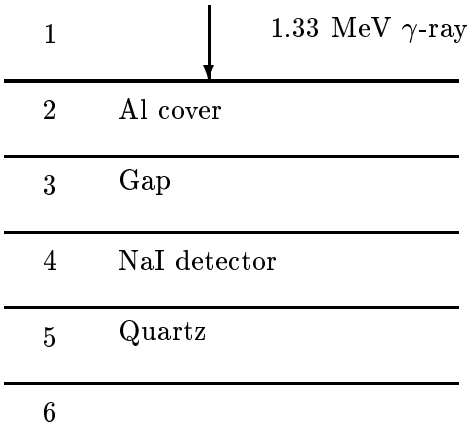


Fig. 2 Semi-infinite slab geometry

Consider a NaI detector comprising the semi-infinite planes shown in Fig.2. There is a 1mm-thick aluminum cover in front of the detector, a 5mm gap between the cover and the detector and a 5mm-thick quartz window after the detector.

All planes are perpendicular to the Z-axis, and are defined as follows:

```
"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;   "Thickness of Al case in cm "
TGAP=0.5;   "Gap between case and detector in cm"
TDE=7.62;   "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"

"DEFINITION OF PLANES"
"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0;   PNORM(2,J)=0.0;   PNORM(3,J)=1.0;
]

"NOW PUT IN THE EXCEPTIONS"

PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;
```

Corresponding HOWFAR is:

```
SUBROUTINE HOWFAR;
;COMIN/DEBUG,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.EQ.NREG) [IDISC=1; RETURN;]

NPL1=IRL; NPL2=IRL-1;
NRG1=IRL+1;
NRG2=IRL-1;

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);
```

```
RETURN;
END;  "END OF SUBROUTINE HOWFAR"
```

As the sample User Codes of a semi-infinite geometry, ucnai1.mor and ucnai1p.mor are distributed from KEK. ucnai1p.mor is a version which uses PRESTA.

3.1.2 Finite Plane Geometry

The finite plane geometry shown in Fig.3 can be treated by defining an additional 4 planes. The outside of these planes is set to region "7".

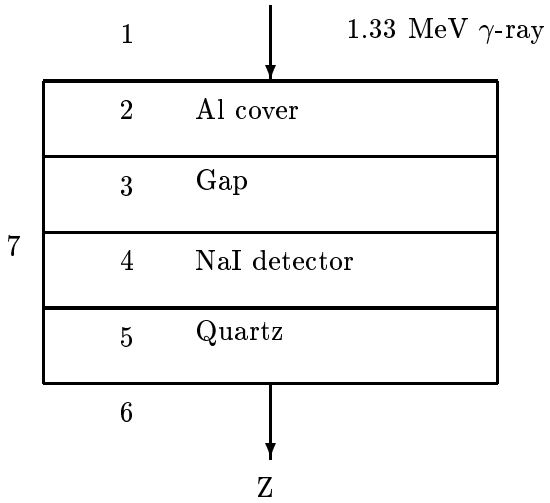


Fig. 3 Finite plane geometry

The planes are defined as follows:

```
"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;    "Thickness of Al case in cm "
TGAP=0.5;    "Gap between case and detector in cm"
TDE=7.62;    "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"
XHALF=3.81; "Half width, X-direction in cm"
YHALF=3.81; "Half width, Y-direction in cm"

"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0;   PNORM(2,J)=0.0;   PNORM(3,J)=0.0;
]

"NOW PUT IN THE EXCEPTIONS"

"Z-direction"

/PNORM(3,1),PNORM(3,2),PNORM(3,3),PNORM(3,4),PNORM(3,5)/=1.0;
PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

"Y-direction"
/PNORM(2,6),PNORM(2,7)/=1.0;
PCOORD(2,6)=-YHALF;
PCOORD(2,7)=YHALF;

"X-direction"
/PNORM(1,8),PNORM(1,9)/=1.0;
PCOORD(1,8)=-XHALF;
PCOORD(1,9)=XHALF;
```

In this example, PCOORD of planes perpendicular to X- or Y-axis are give by the half width (XHALF or YHALF) of each plane.

HOWFAR of this geometry is :

```
SUBROUTINE HOWFAR;
;COMIN/DEBUG,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.NREG-1) [IDISC=1; RETURN;]

NPL1=IRL; NPL2=IRL-1;
NRG1=IRL+1;
NRG2=IRL-1;

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

$PLAN2P(7,7,1,6,7,-1);

$PLAN2P(9,7,1,8,7,-1);
```

Changes from

```
IF(IRL.EQ.1.OR.IRL.EQ.NREG) [IDISC=1;RETURN;]
```

to

```
IF(IRL.EQ.1.OR.IRL.GE.NREG-1) [IDISC=1;RETURN;]
```

is necessary due to an increase in the discard region.

(See ucnai2.mor or ucnai2p.mor as examples of this geometry.)

3.1.3 A 3-Dimensional Cartesian Geometry

It is also possible to apply $PLAN2P for 3-dimensional Cartesian geometry[4].

The geometrical structure for this geometry is divided into a number of bins in the X-, Y- and Z-directions.

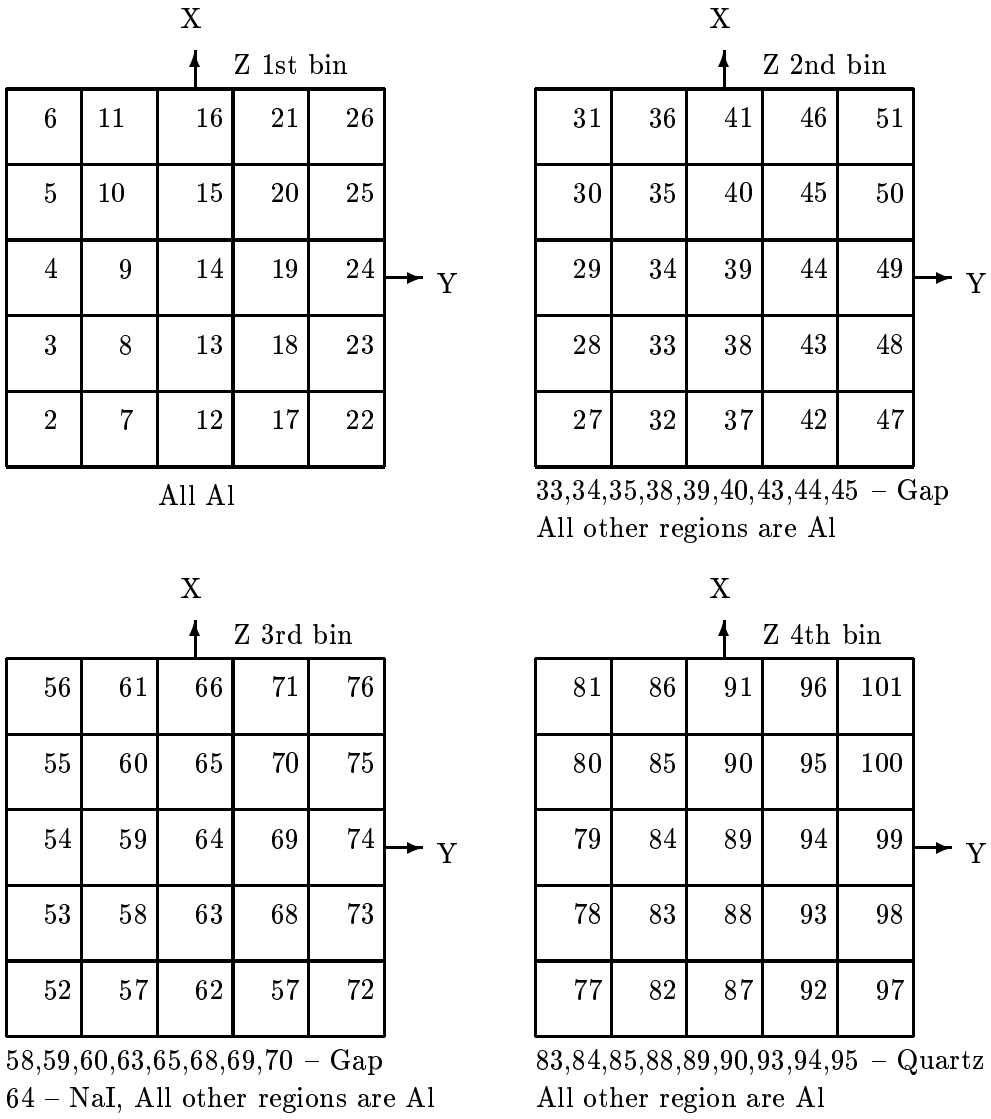


Fig. 4. 3-dimensional Cartesian geometry

The number of bins in each case is defined by `PARAMETER` macro.

```
PARAMETER $NXBIN=5;  "NUMBER OF BINS IN X-DIRECTION"
PARAMETER $NYBIN=5;  "NUMBER OF BINS IN Y-DIRECTION"
"$NXBIN and $NYBIN must be odd"
PARAMETER $NZBIN=4;  "NUMBER OF BINS IN Z-DIRECTION"
PARAMETER $MXREG=106; "Number of region"
```

The total number of regions inside the rectangular parallelepiped is defined as `IXYZ`(200 in this example) and the number of regions in a slab as `IXY` (25 in this example):

```
NPLAN=$NXBIN+$NYBIN+$NZBIN+3;  "ACTUAL NUMBER OF PLANES"
IXY=$NXBIN*$NYBIN;  IXYZ=IXY*$NZBIN;
```

The total number of regions into which the space is divided is in fact `IXYZ+6`.

```
NREG=IXYZ+6;  "ACTUAL NUMBER OF REGION USED"
```

The way in which the region is numbered is illustrated in Fig. 4.

```
"SET MEDIUM INDEX FOR EACH REGION"
DO I=1,NREG [MED(I)=0;]  "Set all region to vacuum at first"

DO IZP=1,$NZBIN [
DO IYP=1,$NYBIN [
DO IXP=1,$NXBIN [
IIZP=(IZP-1)*IXY+(IYP-1)*$NXBIN+IXP+1;
IF(IZP.EQ.1) [MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]
ELSEIF(IZP.EQ.$NZBIN) [MED(IIZP)=2; "set to Al at first"
IF((IXP.GT.1.AND.IXP.LT.$NXBIN).AND.(IYP.GT.1.AND.IYP.LT.$NYBIN)) [
MED(IIZP)=3;  "Quartz region"]
ECUT(IIZP)=0.561;]
ELSEIF(IZP.EQ.2) [
IF(IYP.EQ.1.OR.IYP.EQ.$NYBIN.OR.(IXP.EQ.1.OR.IXP.EQ.$NXBIN)) [
MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]]
ELSE [
IF(IYP.EQ.1.OR.IYP.EQ.$NYBIN.OR.(IXP.EQ.1.OR.IXP.EQ.$NXBIN)) [
MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]
ELSE [
IF((IYP.GE.3.AND.IYP.LE.$NYBIN-2).AND.
(IXP.GE.3.AND.IXP.LE.$NXBIN-2)) [
MED(IIZP)=1;   "NaI(Tl) detector region"
IEDGFL(IIZP)=53;  "53:Atomic number of I"
"                 0:K-X ray of I is not produced"
]]]
]]]
```

The coordinates and normals to the planes should be defined:

```
"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;   "Thickness of Al case in cm "
TGAP=0.5;   "Gap between case and detector in cm"
TDE=7.62;   "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"
XWIDTH=7.62; "Width, X-direction in cm"
YWIDTH=7.62; "Width, Y-direction in cm"
"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0;   PNORM(2,J)=0.0;   PNORM(3,J)=0.0;
]

ID1=$NZBIN+1;  ID2=ID1+1;  ID3=ID2+$NYBIN;
ID4=ID3+1;     ID5=ID4+$NXBIN;
DO I=1,ID1 [PNORM(3,I)=1.0;]
DO I=ID2,ID3 [PNORM(2,I)=1.0;]
DO I=ID4,ID5 [PNORM(1,I)=1.0;]

"NOW PUT IN THE EXCEPTIONS"

"Z-direction"

/PNORM(3,1),PNORM(3,2),PNORM(3,3),PNORM(3,4),PNORM(3,5)/=1.0;
PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
```

```
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

"Y-direction"

PCOORD(2,ID2)=-TCOV-TGAP-0.5*YWIDTH;
PCOORD(2,ID2+1)=PCOORD(2,ID2)+TCOV;
PCOORD(2,ID2+2)=PCOORD(2,ID2+1)+TGAP;
PCOORD(2,ID2+3)=PCOORD(2,ID2+2)+YWIDTH;
PCOORD(2,ID2+4)=PCOORD(2,ID2+3)+TGAP;
PCOORD(2,ID2+5)=PCOORD(2,ID2+4)+TCOV;

"X-direction"
PCOORD(1,ID4)=-TCOV-TGAP-0.5*XWIDTH;
PCOORD(1,ID4+1)=PCOORD(1,ID4)+TCOV;
PCOORD(1,ID4+2)=PCOORD(1,ID4+1)+TGAP;
PCOORD(1,ID4+3)=PCOORD(1,ID4+2)+XWIDTH;
PCOORD(1,ID4+4)=PCOORD(1,ID4+3)+TGAP;
PCOORD(1,ID4+5)=PCOORD(1,ID4+4)+TCOV;
```

The plane number must be assigned in the order Z, Y, and X.
`SUBROUTINE HOWFAR`, which handles the geometry defined above, is:

```
SUBROUTINE HOWFAR;
;COMIN/DEBUG,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN,IXY,IXYZ;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.EQ.1.OR.IRL.GE.IXYZ+2) [IDISC=1; RETURN;]

I=(IRL-2)/IXY+1;  "SLAB NUMBER"
J=(IRL-2-IXY*(I-1))/$NXBIN+1;   "COLUMN NUMBER"
K=IRL-1-IXY*(I-1)-$NXBIN*(J-1); "ROW NUMBER"

NPL1=I+1;  NPL2=I;
IF(I.LT.$NZBIN) [NRG1=IRL+IXY;] ELSE [NRG1=IXYZ+2;]
IF(I.GT.1) [NRG2=IRL-IXY;] ELSE [NRG2=1;]
$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

NPL2=$NZBIN+1+J;  NPL1=NPL2+1;
IF(J.LT.$NYBIN) [NRG1=IRL+$NXBIN;] ELSE [NRG1=IXYZ+3;]
IF(J.GT.1) [NRG2=IRL-$NXBIN;] ELSE [NRG2=IXYZ+4;]
$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

NPL2=$NZBIN+$NYBIN+2+K; NPL1=NPL2+1;
IF(K.LT.$NXBIN) [NRG1=IRL+1;] ELSE [NRG1=IXYZ+5;]
IF(K.GT.1) [NRG2=IRL-1;] ELSE [NRG2=IXYZ+6;]
$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

RETURN;
END;
```

In this subroutine, the shortest length to the six planes surrounding the particle is compared with `USTEP`.

(See ucnai4.mor or ucnai4p.mor as examples of 3-dimensional Cartesian geometry)

4 The $CYLNDR and $CYL2 Macros

The $CYLNDR macro treats a cylinder which has the Z-axis as symmetry axis is:

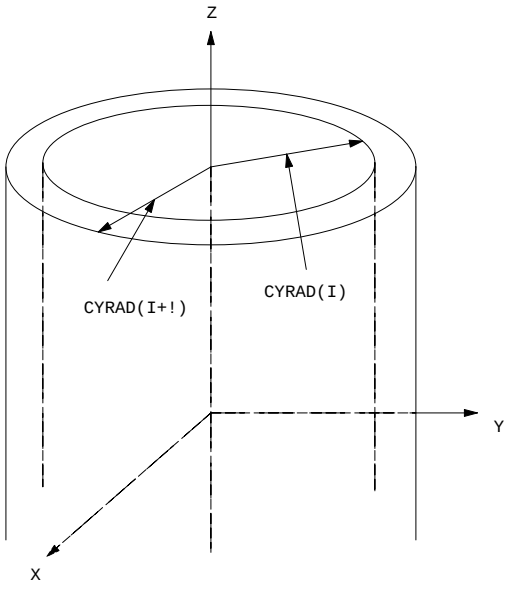
```
$CYLNDR(ICYL,ISIDE,IHIT,TCYL);
    ICYL: ID number of cylinder to be checked.
   ISIDE: 1 means particle is inside cylinder.
          0 means particle is outside cylinder.
    IHIT: 1 means particle intersects surface.
          0 means particle misses surface.
    TCYL: distance to surface if IHIT=1.
```

CYRAD(I) and CYRAD2(I) must be defined to use $CYLNDR or $CYL2. CYRAD is the radius of the cylinder and CYRAD2 is its square. They are included in COMMON/CYLDTA;.

The macro corresponding to $PLAN2P for the plane is $CYL2, which treats particles between two cylinders.

```
$CYL2(NCY1,NRG1,NCY2,NRG2);
    NCY1: ID number of first cylinder to be checked.
          particle must be outside the first cylinder.
    NRG1: region to go into if first cylinder is intersected by particle.
    NCY2: ID number of second cylinder to be checked.
          particle must be inside the second cylinder.
    NRG2: region to go into if second cylinder is intersected by particle.
```

The $CYL2 macro is written to check the cylinder inside the particle first, and then to check the cylinder outside the particle.



CYRAD2(I)=CYRAD(I)*CYRAD(I);

Fig. 5

4.1 Cylinder Slab Geometry

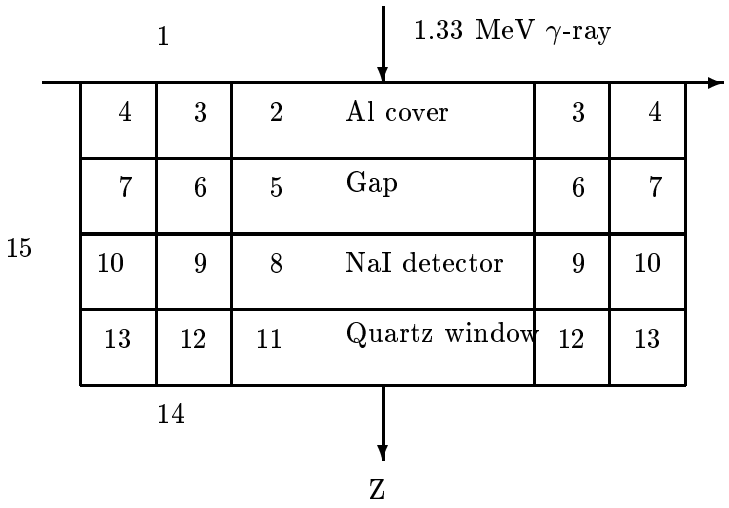


Fig. 6 Cylinder slab geometry

2,3,4,7,10,13 – Al

11,12 – Quartz

5,6,9 – Gap

8 – NaI

The cylinder slab geometry which is defined by the combination of cylinders and planes is the most commonly used geometry.

Consider the geometry defined by 5 planes (Z-direction) and 3 cylinders, shown in Fig. 6. The number of planes and cylinders is defined by:

```
PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
```

Material is assigned to each region as follows:

```
NPLAN=$MXPLNS;    "NUMBER OF PLANES"
NCYL=$MXCYLS;    "NUMBER OF CYLINDER"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
"                       REGION)                                    "
IRZ=NREG-3;
"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0;   "VACUUM REGIONS"
/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
"                                    Al region"
/MED(11),MED(12)/=3;  "Quartz window region"
MED(8)=1;   "NaI(Tl) detector region"

/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0;  "Vacuum region inside case"
```

The cylinders in Fig 6 are defined as follows:

```
"DEFINE THE CYLINDER RADII"

RDET=3.81;  "Radius of detector in cm"
RGAP=0.5;   "Gap between detector and case in cm"
RTCOV=0.1;  "Cover thickness in cm"

CYRAD(1)=RDET;
CYRAD(2)=CYRAD(1)+RGAP;
CYRAD(3)=CYRAD(2)+RTCOV;

"PRINT OUT THE CYLINDER RADII"
OUTPUT ;(///,' CYLINDER RADII',/);
```

```
DO I=1,NCYL [
CYRAD2(I)=CYRAD(I)*CYRAD(I);
OUTPUT I,CYRAD(I);('  I=',I5,5X,' CYRAD(I)=',G15.5,'CM');
]
```

SUBROUTINE HOWFAR which handle this geometry is:

```
SUBROUTINE HOWFAR;
;COMIN/DEBUG,CYLDTA,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN,NCYL,IRZ;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]

RETURN;
END;
```

(See ucnai3.mor or ucnai3p.mor as examples of the cylinder slab geometry)

4.2 The DELCYL Parameter

The algorithm for $CYLNDR contains a parameter called DELCYL, which represents the closest allowable distance of the trajectory to the quadratic surface.

For any given machine precision, the need for DELCYL arises because the quadratic solution, together with the way EGS uses it, can result in a particle being stepped sideways to the surface.

Although this side-stepping is generally small, it can confuse SUBROUTINE HOWFAR to the extent that the code can become hung up in an infinite loop.

The problem has been addressed by Stevenson (CERN), who is responsible for the DELCYL addition to $CYLNDR, and has recommended using a value of 1.0E-4 cm for most problems.

DELCYL is implemented by means of the $DELCYL macro in EGS4MAC (1.E-4 default).

At SLAC, $DELCYL is replaced by the variable DELCYL to dynamically change within HOWFAR, as follows:

```
    IF(CYRAD(IRL).LT.0.1) [DELCYL=1.E-6;]
ELSEIF(CYRAD(IRL).LT.1.0) [DELCYL=1.E-5;]
ELSEIF(CYRAD(IRL).LT.10.) [DELCYL=1.E-4;]
                      ELSE [DELCYL=1.E-3]
```

$CYLNDR is modified to include the SLAC procedure at KEK, and is included in kekmac.mor.

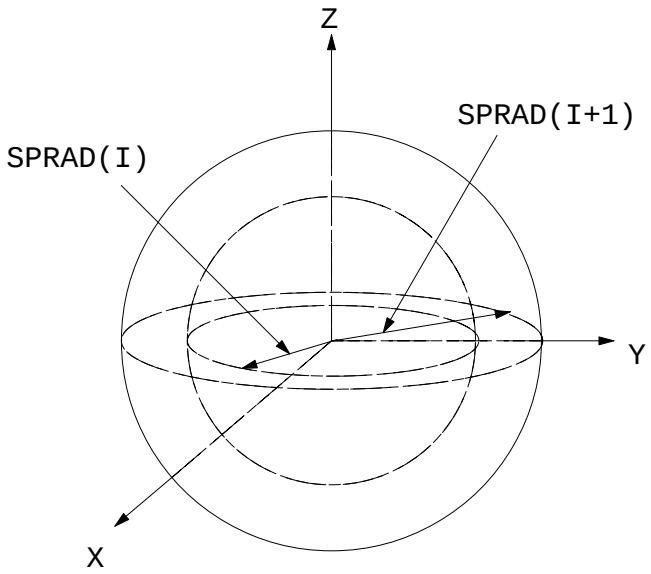
5 The $SPHERE and $SPH2 Macros

The macro which treats a sphere of which the center is the origin is:

```
$SPHERE(ISPH,ISIDE,IHIT,TSPH);
    ISPH: ID number of sphere to be checked.
   ISIDE: 1 means particle is inside sphere.
          0 means particle is outside sphere.
    IHIT: 1 means particle intersects surface.
          0 means particle misses surface.
    TSPH: distance to surface if IHIT=1.

$SPH2(NSPH1,NRG1,NSPH2,NRG2);
    NCY1: ID number of first sphere to be checked.
          particle must be outside the first sphere.
    NRG1: region to go into if first sphere is intersected by particle.
    NCY2: ID number of second sphere to be checked.
          particle must be inside the second sphere.
    NRG2: region to go into if second sphere is intersected by particle.
```

SPRAD(I) and SPRAD2(I) must be defined to use $SPHERE or $SPH2. SPRAD is the radius of the sphere and SPRAD2 is its square. They are included in COMMON/SPHDTA;.



SPRAD2(I)=R(I)*R(I);

6 The $CONE, $CON2 and $CON21 Macros

A cone is characterized by an angle and a displacement. The cone considered here has the Z-axis as the symmetry axis, and, thus, the angle is the angle to the Z-axis. The displacement SMALLL is the Z-coordinate of the vertex.

COTAL(I)=cot(θ_i) and COTAL2(I)=COTAL(I)*COTAL(I) are used to provide information about the angle, where θ is the opening angle to Z-axis, as shown in Fig.6. These variables are included in COMMON/CONDTA.

```
$CONE(ICONE,ISIDE,IHIT,TCONE);
   ICONE: ID number of cone to be checked.
   ISIDE: 1 means particle is inside cone.
          0 means particle is outside cone.
    IHIT: 1 means particle intersects surface.
          0 means particle misses surface.
   TCONE: distance to surface if IHIT=1.
```

When a particle is between 2 cones, the following macros, which correspond to `$PLAN2P` for a plane, are useful:

```
$CON2(NCON1,NRG1,NCON2,NRG2);
$CON21(NCON1,NRG1,NCON2,NRG2);
   NCON1: ID number of first cone to be checked.
          particle must be outside the first cone.
    NRG1: region to go into if first cone is intersected by particle.
   NCON2: ID number of second cone to be checked.
          particle must be inside the second cone for $CON2.
          particle must be outside the second cone for $CON21.
    NRG2: region to go into if second cone is intersected by particle.
```

`$CON2` and `$CON21` are used for the particle positioned at the point (a) and (b) in Fig. 6, respectively.

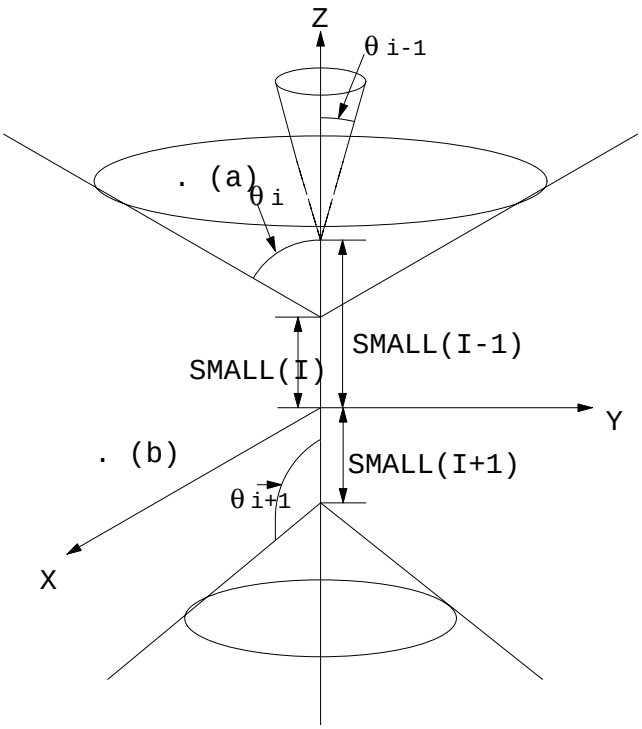


Fig. 6

7 The `$FINVAL` Macro

The `$FINVAL` macro is useful for determining the final coordinates of a particle.

```
$FINVAL(DIST,XCOORD,YCOORD,ZCOORD);
   DIST  : the distance traveled.
   XCOORD: X-coordinate after travel.
   YCOORD: Y-coordinate after travel.
   ZCOORD: Z-coordinate after travel.
```

8 Modification of Geometry Macros

It is possible to modify the geometry macros included in the EGS4 system. As an examples, a cylinder having its symmetry axis parallel to the Z-axis can be handled by the modifications given below.

Original `$CYLNDR` is:

```
REPLACE {$CYLNDR(#,#,#,#);} WITH
   {;IF(CYRAD({P1}).LT.0.1) [DELCYL=1.E-6;]
   ELSEIF(CYRAD({P1}).LT.1.0) [DELCYL=1.E-5;]
   ELSEIF(CYRAD({P1}).LT.10.) [DELCYL=1.E-4;]
   ELSE [DELCYL=1.E-3;]
   NOFCY={P2};{P3}=1;{P4}=0.0;ACYL=SQRT(U(NP)*U(NP)+V(NP)*V(NP));
   IF(ACYL.EQ.0.0)[{P3}=0;]  ELSE [
   BCYL=(X(NP)*U(NP)+Y(NP)*V(NP))/ACYL;CCYL=X(NP)*X(NP)+Y(NP)*Y(NP)
   -CYRAD2({P1});  ARGCY=BCYL*BCYL-CCYL;
   IF(ARGCY.LT.0.0) [{P3}=0;]   ELSE [
   IF(ABS(CCYL).LT.DELCYL.AND.NOFCY.EQ.0.AND.BCYL.GE.0.0)[{P3}=0;]
   ELSEIF(ABS(CCYL).LT.DELCYL.AND.NOFCY.EQ.1.AND.BCYL.LT.0.0)[
   {P4}=-2.0*BCYL/ACYL;]
   ELSE [IF(NOFCY.EQ.1.AND.CCYL.GE.0.0)[{P3}=1;{P4}=DELCYL;]
   ELSEIF(NOFCY.EQ.0.AND.CCYL.LE.0.0)[{P3}=1;{P4}=DELCYL;]
   ELSE  [ROOTCY=SQRT(ARGCY); IF(CCYL.LT.0.0)
   [{P4}=(-BCYL+ROOTCY)/ACYL;]
   ELSEIF(BCYL.LT.0.0) [{P4}=(-BCYL-ROOTCY)/ACYL;]
   ELSE [{P3}=0;]]]]]]}
```

New variables (`XOP, YOP`), which are the X- and Y-coordinates of the symmetry axis, are introduced and `$CYLNDR` is modified to:

```
REPLACE {$CYLNDR(#,#,#,#);} WITH
   {;IF(CYRAD({P1}).LT.0.1) [DELCYL=1.E-6;]
   ELSEIF(CYRAD({P1}).LT.1.0) [DELCYL=1.E-5;]
   ELSEIF(CYRAD({P1}).LT.10.) [DELCYL=1.E-4;]
   ELSE [DELCYL=1.E-3;]
   NOFCY={P2};{P3}=1;{P4}=0.0;ACYL=SQRT(U(NP)*U(NP)+V(NP)*V(NP));
   XNP=X(NP)-XOP({P1});YNP=Y(NP)-YOP({P1});
   IF(ACYL.EQ.0.0)[{P3}=0;]  ELSE [
   BCYL=(XNP*U(NP)+YNP*V(NP))/ACYL;CCYL=XNP*XNP+YNP*YNP
   -CYRAD2({P1});  ARGCY=BCYL*BCYL-CCYL;
   IF(ARGCY.LT.0.0) [{P3}=0;]   ELSE [
   IF(ABS(CCYL).LT.DELCYL.AND.{P2}.EQ.0.AND.BCYL.GE.0.0)[{P3}=0;]
   ELSEIF(ABS(CCYL).LT.DELCYL.AND.{P2}.EQ.1.AND.BCYL.LT.0.0)[
   {P4}=-2.0*BCYL/ACYL;]
   ELSE [IF(NOFCY.EQ.1.AND.CCYL.GE.0.0)[{P3}=1;{P4}=DELCYL;]
   ELSEIF(NOFCY.EQ.0.AND.CCYL.LE.0.0)[{P3}=1;{P4}=DELCYL;]
   ELSE  [ROOTCY=SQRT(ARGCY); IF(CCYL.LT.0.0)
   [{P4}=(-BCYL+ROOTCY)/ACYL;]
   ELSEIF(BCYL.LT.0.0) [{P4}=(-BCYL-ROOTCY)/ACYL;]
   ELSE [{P3}=0;]]]]]]}
```

`COMIN/CYLDTA` is also must be modified to include `XOP,YOP`.

```
"CYLDTA---COMMON BLOCK FOR $CYLNDR MACRO"
REPLACE {;COMIN/CYLDTA/;} WITH {;COMMON/CYLDTA/CYRAD2($MXCYLS),
         CYRAD($MXCYLS),XOP($MXCYLS),YOP($MXCYLS);}
```

9 Modification of Standard `HOWFAR`

It happens sometimes that almost all parts of the geometry can be handled with the standard `HOWFAR` shown above, like a cylinder-slab geometry, though limited parts have different geometries like in Fig. 7. In Fig.7, a cone must be treated as if a particle is positioned inside region 3 or 4. In this case, `HOWFAR` is modified to treat a particle in region 3 or 4 separately.

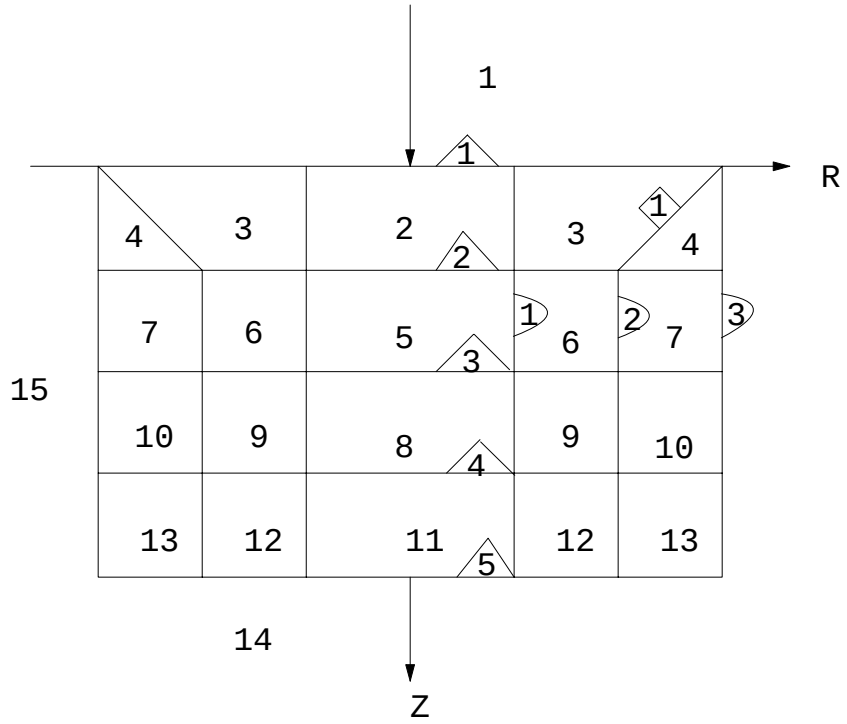


Fig. 7 Cylinder-Slab geometry including a cone

```
IRL=IR(NP);  "SET LOCAL VARIABLE
IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]
NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(IRL.NE.3.AND.IRL.NE.4) [
IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]
]

ELSEIF(IRL.EQ.3) [
$CYLNDR(NANNU-1,0,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,IRL-1);]
$CONE(1,1,IHIT,TCONE);
IF(IHIT.EQ.1) [
$CHGTR(TCONE,4);]
]

ELSE [
$CYLNDR(NANNU,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,IRZ+3);]
$CONE(1,0,IHIT,TCONE);
IF(IHIT.EQ.1) [
$CHGTR(TCONE,3);]
```

```
]

RETURN;
END;
```

References

[1] W. R. Nelson and T. M. Jenkins,"Writing `SUBROUTINE HOEFAR` for EGS4", *SLAC-TN-87-4*, 31 August 1988/Rev.

[2] W. R. Nelson and T. M. Jenkins,"Geometry Methods and Packages", Chapter 17 in *Monte Carlo Transport of Electron and Photons* Plenum Press, 1988.

[3] W. R. Nelson,"HOEFAR:How to Code Geometry", EGS4 Lecture Note.

[4] G. R. Stevenson,"A 3-Dimensional Cartesian Geometry Package for HOWFAR in the EGS Electron Gamma Shower Program", *HS-RP/TM/80-60*.

How to Write Source Routine

1 Sampling Method

1.1 Direct method

A probability distribution function (PDF:$f(x)$) is defined over the range $[a, b]$, where neither a nor b is necessary finite. A PDF must have the properties that it is both integrable and non-negative.

We now construct its cumulative probability function (CDF:$F(x)$),

$$\varsigma = F(x) = \int_a^x f(x)dx, \tag{1}$$

and assume that it is properly normalized, *i.e.* $F(b) = 1$.

By its definition, we can map $F(x)$ onto a range of random variables, ς, where $0 \geq \varsigma \leq 1$.

Having mapped the random numbers onto $F(x)$, we may invert the equation to give

$$x = F^{-1}(\varsigma). \tag{2}$$

In the other case, x takes on discrete values (x_i) with probabilities (p_i) such that

$$F(x_N) = \sum_{i=1}^{N} p_i = 1. \tag{3}$$

In this case, $x = x_i$ if

$$F(x_{i-1}) = \sum_{j=1}^{i-1} p_j \leq \varsigma < F(x_i) = \sum_{j=1}^{i} p_j. \tag{4}$$

The sampling method which can determine x directly from PDF is called a direct method.

1.2 Combination of "composition" and "rejection" techniques

When x has the density function f given by

$$f(x) = \sum_{i=1}^{n} \alpha_i f_i(x) g_i(x), \quad (\alpha_i : \text{positive real number}) \tag{5}$$

$$\int_a^b f_i(x)dx = 1, \tag{6}$$

$$0 \leq g_i(x) \leq 1.0 : \tag{7}$$

1. Pick ς_1 and let i be such that

$$\sum_{j=1}^{i-1} \alpha_j < \varsigma_1 \sum_{j=1}^{n} \alpha_j \leq \sum_{j=1}^{i} \alpha_j. \tag{8}$$

2. Pick x from $f_i(x)$, possibly by solving

$$\int_a^x f_i(x)dx = \varsigma_2. \tag{9}$$

3. Pick ς_3. Terminate the algorithm and accept the value of x if

$$\varsigma_3 < g_i(x). \tag{10}$$

4. Otherwise, go back to step 1.

This sampling method is called a combination of the "composition" and "rejection" techniques.

If $G_i = 1$ for all i, it is called the "composition" method. If $N = 1$, it is called the"rejection" method.

1.3 von Neumann's method

In some cases, von Neumann's method, which is a kind of "rejection" method, is useful to determine x.

This is the case that f_1 and $g(x)$ are supposed is defined as

$$f_1(x) = \frac{1}{b-a} \tag{11}$$

$$, g(x) = y = f^*(x) = f(x)/(\text{Maximum value of } f(x)). \tag{12}$$

1. Sample point x_i from $f_1(x)$.

$$\xi = a + \varsigma_1(b-a); \varsigma_1 = \int_a^{\xi} dx/(b-a) = (\xi - a)/(b-a) \tag{13}$$

2. Calculate y for ξ, $y = f^*(\xi)$.

3. Pick ς_2. Terminate the algorithm and accept the value of x if

$$\varsigma_2 < y. \tag{14}$$

4. Otherwise, go back to step 1.

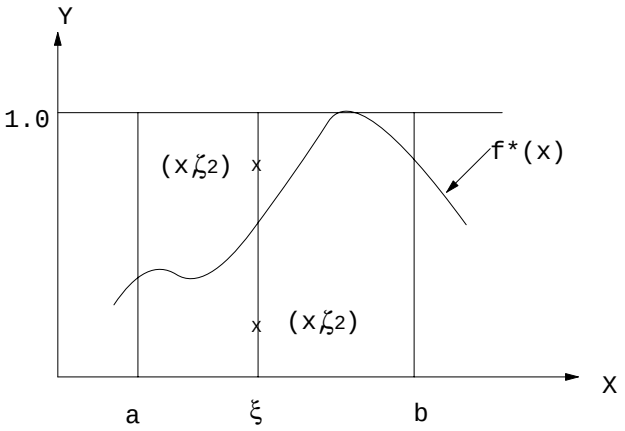


Figure 1: von Neumann's method.

2 Source Routine

The following source routines, which determine the position, direction and energy of a source particle, must be placed before `CALL SHOWER;`.

2.1 Determination of Position

2.1.1 Line source

Suppose a source which is uniformly distributed in the region $a \leq x < b$. In this case, the PDF is

$$f(x)dx = dx/(b-a); \int_a^b f(\xi)d\xi = 1. \tag{15}$$

By solving

$$\varsigma = F(x) = \int_a^x f(x)dx = (x-a)/(b-a) \tag{16}$$

the position x is determined as $x = a + \varsigma(b-a)$.

List of a sampling routine.

```
$RANDOMSET RNO;
XI=XMIN+RNO*(XMAX-XMIN);
```

2.1.2 Uniform source on an annulus of radii $\mathbf{R_0} < \mathbf{R_1}$

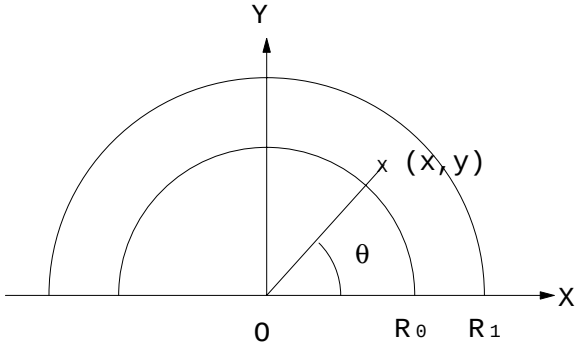


Figure 2: Uniform source on a annulus of radii $R_0 < R_1$.

Suppose a source uniformly distributed inside an annual area between radius R_0 and R_1 in the X-Y plane.

In this case, the PDF for the radius is

$$f(r)dr = 2\pi r dr/\pi(R_1^2 - R_0^2) = 2rdr/(R_1^2 - R_0^2); \int_{R_0}^{R_1} f(r)dr = 1. \tag{17}$$

The radial position is obtained by solving

$$\varsigma_1 = F(r) = \int_{R_0}^{r} f(r)dr = (r^2 - R_0^2)/(R_1^2 - R_0^2) \tag{18}$$

and $r = \sqrt{R_0^2 + \varsigma_1(R_1^2 - R_0^2)}$.

Having thus located radius R, we note that $f(\phi)d\phi = d\phi/2\pi$, so that the next random number determines ϕ by

$$\varsigma_2 = F(\phi) = \int_{-\pi}^{\phi} f(\phi)d\phi = (\phi + \pi)/2\pi \tag{19}$$

and $\phi = \pi(2\varsigma_2 - 1)$.

x and y are calculated from

$$x = r\cos\phi \tag{20}$$

$$, y = r\sin\phi. \tag{21}$$

List of a sampling routine.

```
$RANDOMSET RN1;
RO2=RO*RO; R12=R1*R1;
RR=SQRT(RO2+RN1*(R12-RO2));
$RANDOMSET RN2;
PHAI=PI*(2.0*RN2-1.0);
" COMIN/UPHIOT/ must be included to use PI"
XI=RR*COS(PHAI);
YI=RR*SIN(PHAI);
```

If R_0=0, radial position r is determined by

$$r = R_1\sqrt{\varsigma_1}. \tag{22}$$

A position (x, y) can be determined al follows in the case of the "rejection" method.

In this technique, a point is chosen randomly within the square $-1 \leq x \leq 1; -1 \leq y \leq 1$. If this point lies within a circle with unit radius the point is accepted and the x and y values are scaled by radius R_1.

List of a sampling routine.

```
:SAMPLING:$RANDOMSET RN3;
$RANDOMSET RN4;
XI0=2.0*RN3-1.0;
YI0=2.0*RN4-1.0;
RR=SQRT(XI0*XI0+YI0*YI0);
IF(RR.GT.1.0) [GO TO :SAMPLING:;
XI=R1*XI0; YI=R1*YI0;
```

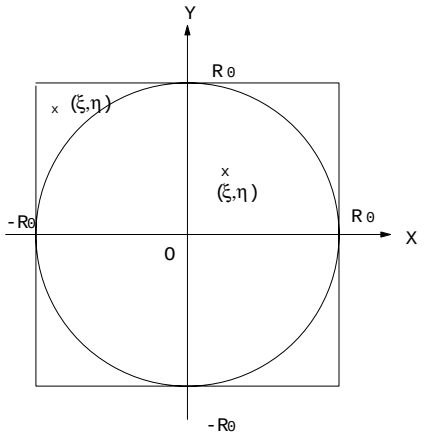


Figure 3: Determines position by the rejection method.

2.1.3 Parallel-beam source incident on a lateral surface of a right-circular cylinder of radius R_1

If the beam is taken in the positive y-direction $(u = 0, v = 1, w = 0)$, and the cylinder in the position shown in Fig.4, one may use the indicated routine.

The first random number determines x by

$$x = R_1(2\varsigma_1 - 1), \tag{23}$$

and y is calculated by

$$y = -\sqrt{(R_1^2 - x^2)}. \tag{24}$$

The second random number determines z by

$$Z = \frac{H}{2}(2\varsigma_2 - 1). \tag{25}$$

List of a sampling routine.

```
$RANDOMSET RN5;
XI=R1*(2.0*RN5-1.0);
YI=-SQRT(R1*R1-XI*XI);
$RANDOMSET RN6;
ZI=H/2.0*(2.0*RN6-1.0);
```

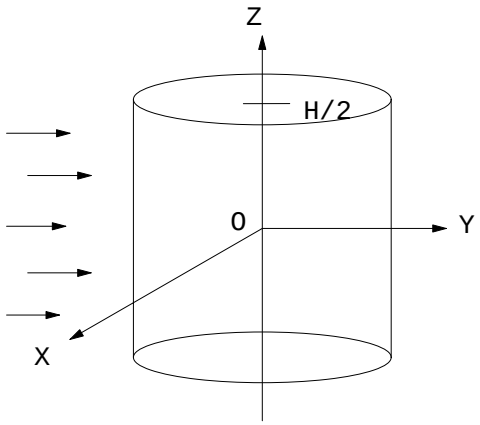


Figure 4: Paralle-beam source incident on lateral surface on cylinder.

2.1.4 Uniform source in a right-circular cylinder of radius R_1

Suppose a uniformly distributed source inside a right-circular cylinder (Fig. 4).
x and y can be determined in the same way as for a uniform circle of radius R_1:

$$r = R_1\sqrt{\varsigma_1} \tag{26}$$
$$\phi = \pi(2\varsigma_2 - 1) \tag{27}$$
$$x = r\cos\phi \tag{28}$$
$$y = r\sin\phi \tag{29}$$

z is determined in the same way as above mentioned.

$$Z = \frac{H}{2}(2\varsigma_3 - 1). \tag{30}$$

List of a sampling routine.

```
$RANDOMSET RN1;
R12=R1*R1;
RR=SQRT(RN1*R12);
$RANDOMSET RN2;
PHAI=PI*(2.0*RN2-1.0);
" COMIN/UPHIOT/ must be included to use PI"
XI=RR*COS(PHAI);
YI=RR*SIN(PHAI);
$RANDOMSET RN3;
ZI=H/2.0*(2.0*RN3-1.0);
```

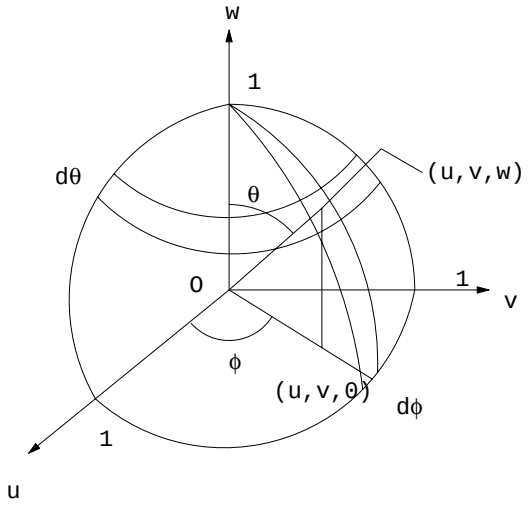


Figure 5: Determination of directional cosine.

2.2 Direction Co-ordinate for Source Particles

2.2.1 Isotropic source

The present problem is tantamount to that of choosing a point (u, v, w) uniformly distributed on the unit sphere $U^2 + v^2 + w^2 = 1$. The element of area for this sphere has spherical coordinates θ, ϕ. The PDF $f(w)$ is given by

$$f(w) = -2\pi \sin\theta d\theta / 4\pi = \frac{1}{2} dw; \int_{-1}^{1} f(w) dw = 1. \tag{31}$$

We may therefore first determine w by

$$\varsigma_1 = \int_{-1}^{w} f(w) dw = \int_{-1}^{w} \frac{1}{2} dw = \frac{1}{2}(w+1) \tag{32}$$

and ϕ subsequently by

$$\phi = \pi(2\varsigma_2 - 1). \tag{33}$$

u and v are given by

$$\sin\theta = \sqrt{1 - w^2} \tag{34}$$

$$, u = \sin\theta \cos\phi \tag{35}$$

$$, v = \sin\theta \cos\phi. \tag{36}$$

List of a sampling routine.

```
$RANDOMSET RN7;
WI=2.0*RN7-1.0;
$RANDOMSET RN8;
PHAI=PI*(2.0*RN8-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;
```

A rejection method can also be applied to this case.

A point (x_i, y_i, z_i) is chosen randomly within the box $-1 \leq x \leq 1.0; -1 \leq y \leq 1.0; -1 \leq z \leq 1.0$. If this point lines within a sphere with unit radius,

$$R = \sqrt{x_1^2 + y_1^2 + z_1^2} \leq 1, \tag{37}$$

u, v, w are determines by

$$u = x_1/R; \quad v = y_1/R; \quad w = z_1/R. \tag{38}$$

Otherwise, sample a point again.

List of a sampling routine.

```
:SAMPLING:$RANDOMSET RN9;
$RANDOMSET RN10;
$RANDOMSET RN11;
X0=2.0*RN9-1.0;
Y0=2.0*RN10-1.0;
Z0=2.0*RN11-1.0;
R0=SQRT(X0*X0+Y0*Y0+Z0*Z0);
IF(R0.GT.1.0) [GO TO :SAMPLING:;]
UI=X0/R0; UI=Y0/R0; WI=Z0/R0;
```

2.2.2 Isotropic point source at $z = -z_0$

If an isotropic point source exits at $z = -z_0$ for the cylinder/plane geometry of `ucnai3.mor` and `ucnai3p.mor`, it is efficient to sample the direction co-ordinates within a limitted θ and to correct any contributions of particles which do not enter the detector with a "weight".

The weight is given by

$$WT = \int_{\theta_0}^{0} (-2\pi \sin\theta d\theta)/4\pi = (1 - \cos\theta_0)/2. \tag{39}$$

In this case, the PDF for $w = \cos\theta$ is

$$f(w)dw = -2\pi \sin\theta d\theta/(2\pi(1 - \cos\theta_0)) = dw/(1 - \cos\theta_0). \tag{40}$$

We determine w by

$$\varsigma_1 = \int_{\cos\theta_0}^{w} f(w)dw = \int_{\cos\theta_0}^{w} \frac{1}{1 - \cos\theta_0} dw = \frac{w - \cos\theta_0}{1 - \cos\theta_0}. \tag{41}$$

If we neglect the interaction between the source position and the detector surface, the direction co-ordinates and positions at the surface of the detector are given by

$$\begin{aligned} WI &= w & (42)\\ ,UI &= \cos\phi\sqrt{1 - w^2} & (43)\\ ,VI &= \sin\phi\sqrt{1 - w^2} & (44)\\ ,XI &= Z0 \times UI & (45)\\ ,YI &= Z0 \times VI & (46)\\ ,ZI &= 0.0. & (47) \end{aligned}$$

List of a sampling routine.

```
"*****************************************************************"
"****** STEP 6.  DETERMINATION OF INCIDENT PARTICLE PROPERTIES ****"
"*****************************************************************"

ZI=0.0;  "STARTING COORDINATES (CM)"
Z0=-10.0;
TT=SQRT(Z0*Z0+CYRAD2(3));
COSMI=Z0/TT;  "Minimum value of WI to enter detector"
WTI=(1-COSMI)/2.0;
.............................

"*****************************************************************"
"******************** STEP 7. SHOWER-CALL---NEXT *****************"
"*****************************************************************"
```

```
DO NOFBAT=1,NBATCH [ "BATCH-LOOP"

DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"

$RANDOMSET WI0;
WI=WI0*(1.0-COSMI)+COSMI;
$RANDOMSET PHAI0;
PHAI=PI*(2.0*PHAI0-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;
XI=ABS(Z0)*UI;
YI=ABS(Z0)*UI;
RR2=XI*XI+YI*YI;
IF(RR2.LE.CYRAD2(1)) [IRI=2;]
ELSEIF(RR2.LE.CYRAD2(2)) [IRI=3;]
ELSE [IRI=4;]

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
  OUTPUT EI,XI,YI,ZI,UI,VI,WI,
  IQI,IRI,IDINC;   (7G15.7,3I5);
  ILINES=ILINES+1;]

CALL SHOWER(IQI,EI,XI,YI,ZI,UI,VI,WI,IRI,WTI);
```

In this example, $UI = \sqrt{1-w^2} and VI = 0.0$ are used due to the symmetry of the geometry to the z-axis. The region that a particle exters is defined from `RR2=XI*XI+YI*YI`.

2.2.3 The cosine distribution

If it is agreed that the outer normal to the surface at the point has the direction $u = 0, v = 0, w = 1$, the cosine distribution based on the definition of PDF, $f(w)$, is

$$f(w) = kwdw; \int_0^1 kwdw = \frac{1}{2}k. \tag{48}$$

k is a constant and is 2 from $\int_0^1 f(w)dw = 1$.

w is determined by

$$\varsigma_1 = \int_0^w 2wdw = w^2. \tag{49}$$

List of a sampling routine.

```
$RANDOMSET RN12;
WI=SQRT(RN12);
$RANDOMSET RN13;
PHAI=PI*(2.0*RN13-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;
```

2.3 Energy of Source Particles

2.3.1 Interpolation method

One usually decides upon a set of energy ranges with lower bounds $E_1 < E_2 < ... < E_n$, with the storage of the physical quantities for each of these ranges, E_n being highest energy permitted to particles in the problem.

If the corresponding CDFs are $F(E_1)$, $F(E_2)$, ..., $F(E_n)$, E is determined by the following procedure:

1. Determine i such that $F(E_i) < \varsigma < F(E_{i+1})$.

2. Apply a linear interpolation between E_i and E_{i+1} to calculate E fron ς by

$$E = E_i + \frac{(\varsigma - F(E_i)) \times (E_{i+1} - E_i)}{F(E_{i+1}) - F(E_i)}. \tag{50}$$

List of a sampling routine.

```
$RANDOMSET RN14;
DO IE=2,NEMAX [
IF(RN14.LE.PDF(IE)) EXIT;
]
EI=ES(IE-1)+(RN14-PDF(IE-1))*(ES(IE)-ES(IE-1))/(PDF(IE)-PDF(IE-1));
```

2.3.2 Rejection method

If the energy spectrum is given by an equation ($p(E)$) and the maximum values of $p(E)$ within the energy range between E_L and E_U, is given by p^*, we can use the rejection method to determine an energy of a particle.

1. $E = E_L + \varsigma_1(E_U - E_L)$
2. If $\varsigma_2 < p(E)/p*$, cite E as the energy of the particle. Otherwise, resample ς_1 and ς_2.

List of a sampling routine.

```
:SAMPLING:$RANDOMSET RN15;
EI=EMIN+RN15*(EMAX-EMIN);
PDF=SPEC(EI)/SPEMAX;
"SPEC is the FUNCTION to calculate spectrum"
$RANDOMSET RN16;
IF(RN16.GT.PDF) [GO TO :SAMPLING:;]
```

2.3.3 Uniform sampling with weight

The previous rejection method reflects the shape of the PDF and, therefore, particles having a small PDF are not sampled frequently.

In some cases, those particles having a small PDF effect the results. In this case, a uniform sampling with weight is effective.

In this method, we sampled E from a uniform distribution between E_L and E_U by

$$E = E_L + \varsigma(E_U - E_L). \tag{51}$$

The weight of the sampled particle is set to $f(E)$.

List of a sampling routine.

```
$RANDOMSET RN17;
EI=EMIN+RN17*(EMAX-EMIN);
WTI=PDF(EI);
"PDF is the probability distribution function of spectrum"
```

Appendix: Lists of ucnai3p.mor

```
!INDENT M3;
!INDENT F2;
"*********************************************************************"
"**************************  STANFORD LINEAR ACCELERATOR CENTER  *"
"***   U C N A I 3 P *******                                        *"
"**************************  EGS4 USER CODE -- 27 JUL 1998/0930  *"
"*********************************************************************"
"                                                                     "
"  PROGRAMMER:   HIDEO HIRAYAMA                                       "
"                NATIONAL LABORATORY FOR HIGH ENERGY PHYSICS (KEK)    "
"                OHO-MACHI, TSUKUBA-GUN, IBARAKI,                     "
"                JAPAN                                                "
"                                                                     "
"*********************************************************************"
"                                                                     "
"  PROGRAM:      UCNAI3                                               "
"                                                                     "
"                EGS4 USER CODE TO CALCULATE NAI RESPONSE             "
"                Cylinder-slab geometry with PRESTA.                  "
"                Add Ranmar random generator option.                  "
"                                                                     "
"*********************************************************************"
"                                                                     "
"                     F E A T U R E S                                 "
"                                                                     "
"    - USES ENERGY CONSERVATION PROGRAM CALLED ECNSV1                 "
"    - USES 'COUNTER' ROUTINE CALLED NTALLY                           "
"                                                                     "
"*********************************************************************"
"                                                                     "
"       THE FOLLOWING 'STEPS' REFER TO THE STEPS OUTLINED             "
"       IN THE EGS3 USER MANUAL (SLAC-210).                           "
"       VARIOUS EGS USER NOTES (EUN'S) HAVE BEEN CREATED              "
"       TO SUPPLEMENT SLAC-210 FOR THE CORRECTIONS, CHANGES           "
"       AND ADDITIONS THAT ARE IN EGS4.                               "
"                                                                     "
"*********************************************************************"
"******** STEP 1.  USER-OVER-RIDE-OF-EGS-MACROS ******************"
"*********************************************************************"
%C80
!NEWCONDITIONAL;
"---------------------------------------------------------------------"
"Select random number generater: 0=RAN6 1=RANMAR                    "
"RANMAR is a Lagged-Fibonacci Method pseudo random number generator"
"devised by George Marsaglia and Arif Zaman.                         "
"---------------------------------------------------------------------"
REPLACE {$RNGEN} WITH {0}

"STEP 1.  USER-OVER-RIDE-OF-EGS-MACROS"

REPLACE {;COMIN/RANDOM/;} WITH {
 {SETR B=$RNGEN}
 [IF] {COPY B}=0 [
   ;COMMON/RANDOM/URNDM(97),IXX,IXXST;
   ]
 [IF] {COPY B}=1 [
   ;COMMON/RANDOM/URNDM(97),CRNDM,CDRNDM,CMRNDM,IXX,JXX;
   ]
   }

REPLACE {$COMMON-RANDOM-DECLARATION-IN-BLOCK-DATA;} WITH {
 {SETR B=$RNGEN}
 [IF] {COPY B}=0 [
   ;COMMON/RANDOM/URNDM(97),IXX,IXXST;
   ]
 [IF] {COPY B}=1 [
   ;COMMON/RANDOM/URNDM(97),CRNDM,CDRNDM,CMRNDM,IXX,JXX;
   ]
```

```
   }

REPLACE {$RANDOMSET#;} WITH {
 {SETR B=$RNGEN}
 [IF] {COPY B}=0 [
   IXX=IXX*663608941;{P1}=IXX*0.23283064E-09;IF(IXX.LT.0){P1}={P1}+1.0;
   IF(IXX.EQ.IXXST) [OUTPUT;(' WARNING !'/
   ' Same random number will be produced.'/
   ' It is better to use RANNMAR random number generator.')]
   ]
 [IF] {COPY B}=1 [
    {P1}=URNDM(IXX)-URNDM(JXX); IF({P1}.LT.0.) {P1}={P1}+1.;
    URNDM(IXX) = {P1};
    IXX=IXX-1; IF(IXX.EQ.0) IXX=97;
    JXX=JXX-1; IF(JXX.EQ.0) JXX=97;
    CRNDM=CRNDM-CDRNDM; IF(CRNDM.LT.0.) CRNDM=CRNDM+CMRNDM;
    {P1}={P1}-CRNDM; IF({P1}.LT.0.) {P1}={P1}+1.;
   ]
   }

"This should be called somewhere near the beginning of the main routine"
"before any random numbers are asked for";
REPLACE {$RNG-INITIALIZATION;} WITH {;
 {SETR B=$RNGEN}
 [IF] {COPY B}=0 [;]
 [IF] {COPY B}=1 [ IXX=0; JXX=0; CALL RMARIN;
      DO II=1,20005[
      $RANDOMSET XRANM;
      IF(II.GT.20000) OUTPUT (MOD(INT(XRANM*16.**JJ),16),JJ=1,7);
      (8X,7I3); ]
    ]
 }

"-------------------------------------------------------------------"
"      PLACE COMPILER DEPENDENT SUBROUTINE CALL HERE                "
"-------------------------------------------------------------------"
" Select Fortran Compiler.                                          "
" Lahey Fortran = 1                                                 "
" Microsoft Fortran = 2                                             "
" Other PC = 3 (default)                                            "
" Sun UNIX Workstation = 4                                          "
" HI-UX Workstation =5                                              "
" Other UNIX Workstation =6                                         "
"-------------------------------------------------------------------"
REPLACE {$COMPILER} WITH {3}
"-------------------------------------------------------------------"

"-------------------------------------------------------------------"
"Macro to select the timer to be used (compiler dependent).         "
"-------------------------------------------------------------------"
REPLACE {$TIMERSET;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=5 [CALL CLOCK(IDUMMY,3);]
  [ELSE]          [;]
}

REPLACE {$TIME-NOW#;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=1 [CALL TIMER({P1});]
  [IF] {COPY B}=2 [CALL GETTIM(IHR,IMIN,ISEC,I100);
                  {P1}=(IHR*3600+IMIN*60+ISEC)*100+I100;]
  [IF] {COPY B}=3 [{P1}=0.0;]
  [IF] {COPY B}=4 [TT=ETIME(TARRAY);
                  {P1}=TARRAY(1)*100.0;]
  [IF] {COPY B}=5 [CALL CLOCK(TT,5);
                  {P1}=TT*100.0;]
  [IF] {COPY B}=6 [{P1}=0.0;]
}

"-------------------------------------------------------------------"
"Macro to define DIMENSION related to the timer to be used          "
"                                    (compiler dependent).          "
"-------------------------------------------------------------------"
```

```
REPLACE {$TIME-DIM;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=4 [REAL TARRAY(2);]
  [ELSE]          [;]
}

"-------------------------------------------------------------------"
"Macro to define OPEN STATEMENT for Cross-section data.            "
"                                          (UNIX or PC).           "
"-------------------------------------------------------------------"
REPLACE {$OPEN;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}<4 [OPEN(12,FILE='mortjob.xse',status='old');
                   OPEN(6,FILE='mortjob.out',status='new');
                   OPEN(8,FILE='mortjob.dum',status='new');]
  [ELSE]          [OPEN(12,FILE='mortjob.xsec',status='old');
                   OPEN(6,FILE='mortjob.output',status='new');
                   OPEN(8,FILE='mortjob.dummy',status='new');]
}

REPLACE{$CALL-HOWNEAR(#);} WITH
 {$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY({P1});}
"                    ****                                          "
"THIS IS THE MACRO THAT SHOULD RETURN THE CLOSEST PERPENDICULAR    "
"DISTANCE TO ANY SURFACE WHICH FORMS A BOUNDARY FOR THE CURRENT    "
"REGION. IN THIS APPLICATION IT IS REPLACED BY THE MACRO FOLLOWING "
"WHICH IS SPECIALIZED FOR THE SLAC       "
"                                                      ****       "
"CYLINDRICAL-PLANE GEOMETRY.                                       "
"IT IS THE USER'S RESPONSIBILITY TO PROVIDE THIS MACRO FOR HIS OWN "
"GEOMETRY.   "
"++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++   "

; "BUFFER FLUSH"
REPLACE{$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY(#);} WITH
"       ==================****========================="
   {;ZL=Z(NP);RL=SQRT(X(NP)**2+Y(NP)**2);
   $GET-JR-JZ(IRL);
   ZLEFT=ZL-PCOORD(3,JZ);ZRIGHT=PCOORD(3,JZ+1)-ZL;ROUT=CYRAD(JR)-RL;
   {P1}=MIN(ZLEFT,ZRIGHT,ROUT);
   IF(JR.NE.1) [RIN=RL-CYRAD(JR-1);{P1}=MIN({P1},RIN);]}
"THIS ROUTINE IS INTENDED TO BE USED TO CALCULATE THE MINIMUM      "
"PERPENDICLAR TO THE NEAREST BOUNDING SURFACE. THIS VERSION IS     "
"SPECIALLY DESIGNED FOR THE SLAC CYLINDRICAL GEOMETRY PACKAGE.     "
"A DIFFERENT VERSION IS NEEDED FOR OTHER    "
"****                                                              "
"GEOMETRY PACKAGES.                                                "

; "BUFFER FLUSH"
"SLAC MACRO TO GET CYLINDER (JR) AND SLAB (JZ) ZONES FROM REGION   "
"NUMBER"
"*****                                                             "
REPLACE {$GET-JR-JZ(#);} WITH
"        =============="
  {;JZ=({P1}-2)/NCYLP+1;JR={P1}-1-NCYLP*(JZ-1);}
; "BUFFER FLUSH"
"GEOMETRICAL INFORMATION"
"SLAC DEFINITION OF /GEOM/ AND RE-DEFINITIONS FOR /CYLDTA/ AND     "
"/PLADTA/        "

"****                                                              "
REPLACE {;COMIN/GEOM/;} WITH {;COMIN/CYLDTA,PLADTA/;}
"       =============="

REPLACE {;COMIN/CYLDTA/;} WITH
"       ================"
  {;COMMON/CYLDTA/CYRAD($MXCYLS),CYRAD2($MXCYLS),NCYLP;}
"NOTE: CYRAD MUST BE PROVIDED IN ADDITION TO CYRAD2 (USUALLY IN    "
"MAIN)"

REPLACE {;COMIN/PLADTA/;} WITH
"       ================"
  {;COMMON/PLADTA/PCOORD(3,$MXPLNS),PNORM(3,$MXPLNS);}
```

```
"COMMON to define variables to score at AUSGAB"
"DEPE:deposited energy inside the detector"
"DELTAE:energy bin width in MeV"
"SPG:Gamma spectrum, SPE:Electron spectrum, SPP:Positron spectrum"

REPLACE {;COMIN/TOTALS/;} WITH
{;COMMON/TOTALS/DEPE,DELTAE,SPG($NDET,$NEBIN),SPE($NDET,$NEBIN),
SPP($NDET,$NEBIN);}

"COMMON of print-out parameter"

REPLACE {;COMIN/LINES/;} WITH
{;COMMON/LINES/NLINES,NWRITE,NCOUNT,ILINES;}

"COMMON of geometry related parameter"

REPLACE {;COMIN/PASSIT/;} WITH
{;COMMON/PASSIT/NREG,NPLAN,NCYL,IRZ;}

PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
PARAMETER $NCASES=5000; "MAXIMUN NUNBER OF CASES"
PARAMETER $NBATCH=50;  "Number of batch"
PARAMETER $NEBIN=50;   "Number of energy bin"
PARAMETER $NDET=1;     "Number of detector"

"*****************************************************************"
"****************** ADDITIONAL (NON-EGS) MACROS ******************"
"*****************************************************************"

"                       N O N E                                   "

"*****************************************************************"
"************************** DECLARATIONS *************************"
"*****************************************************************"

;COMIN/DEBUG,BOUNDS,BREMPR,EDGE,ELECIN,ETALY1,GEOM,LINES,MEDIA,MISC,
NTALY1,PASSIT,RANDOM,STACK,THRESH,TOTALS,UPHIOT,USEFUL,USER/;
DIMENSION PH($NEBIN),PHPB($NEBIN,$NBATCH);
DIMENSION SPGPB($NDET,$NEBIN,$NBATCH),SPEPB($NDET,$NEBIN,$NBATCH),
          SPPPB($NDET,$NEBIN,$NBATCH);
DIMENSION PEFPB($NBATCH),TEFPB($NBATCH);
REAL*8 TOTKE,AVAILE,DEPE;
       "NEEDED FOR ENERGY CONSERVATION TABULATION"

$TYPE   MEDARR(24,3);
DATA MEDARR/$S'NAI-IAPRIM',14*' ',
            $S'AL-IAPRIM',15*' ',
            $S'QUARTZ-IAPRIM',11*' '/;


"*****************************************************************"
"*************** START OF EXECUTABLE CODE ************************"
"*****************************************************************"

$OPEN;

"*****************************************************************"
"******** STEP 2.  PRE-HATCH-CALL-INITIALIZATION COMES NEXT *******"
"*****************************************************************"
NMED=3; "NUMBER OF MEDIA"

DO J=1,NMED [
DO I=1,24 [MEDIA(I,J)=MEDARR(I,J);]]

NPLAN=$MXPLNS;    "NUMBER OF PLANES"
NCYL=$MXCYLS;    "NUMBER OF CYLINDER"
NCYLP=NCYL;   "*PRESTA*"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
"                       REGION)                                   "
IRZ=NREG-3;
```

```
"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0;   "VACUUM REGIONS"

/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
"                                    Al region"
MED(8)=1;   "NaI(Tl) detector region"
/MED(11),MED(12)/=3; "Quartz region"

/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0;  "Vacuum region inside case"

IEDGFL(8)=53;  "53:Atomic number of I"
"               0:K-X ray of I is not produced"

"*********************************************************************"
"***************** STEP 3.  HATCH-CALL COMES NEXT ****************"
"*********************************************************************"

CALL HATCH;

"OUTPUT VARIOUS QUANTITIES ASSOCIATED WITH THE MEDIA"

OUTPUT;  ('1QUANTITIES ASSOCIATED WITH EACH MEDIA:',//);

DO J=1,NMED [
OUTPUT (MEDIA(I,J),I=1,24);  (/,1X,24A1);
OUTPUT RHO(J),RLC(J);  (5X,' RHO=',G15.7,' G/CM**3     RLC=',
   G15.7,' CM');
OUTPUT AE(J),UE(J);  (5X,' AE=',G15.7,' MEV    UE=',G15.7,' MEV');
OUTPUT AP(J),UP(J);  (5X,' AP=',G15.7,' MEV    UP=',G15.7,' MEV');
]

OUTPUT;(' INFORMATION OF MEDIUM AND CUT-OFFFOR EACH REGION'//);
DO I=1,NREG [
IF(MED(I).EQ.0) [OUTPUT I,ECUT(I),PCUT(I);
(' MEDIUM(',I3,')=VACUUM',18X,'ECUT=',G10.5,' MEV, PCUT=',G10.5,' MEV');
]
ELSE [OUTPUT I,(MEDIA(II,MED(I)),II=1,24),ECUT(I),PCUT(I);
(' MEDIUM(',I3,')=',24A1,'ECUT=',G10.5,' MEV, PCUT=',G10.5,' MEV');]
]

DO I=1,NREG [IF(IEDGFL(I).NE.0) [CALL EDGSET(NREG); EXIT;]]

"*********************************************************************"
"*********** STEP 4.  HOWFAR-INITIALIZATION COMES NEXT ***********"
"*********************************************************************"

"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;   "Thickness of Al case in cm "
TGAP=0.5;   "Gap between case and detector in cm"
TDE=7.62;   "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"

"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0;   PNORM(2,J)=0.0;   PNORM(3,J)=1.0;
]

"NOW PUT IN THE EXCEPTIONS"

PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

OUTPUT; ('1PCOORD AND PNORM VALUES FOR EACH J-PLANE (I=1,3):',//);
```

```
DO J=1,NPLAN [
OUTPUT J,(PCOORD(I,J),I=1,3),(PNORM(I,J),I=1,3);
(I5,6G15.7);]

"DEFINE THE CYLINDER RADII"

RDET=3.81;  "Radius of detector in cm"
RGAP=0.5;   "Gap between detector and case in cm"
RTCOV=0.1;  "Cover thickness in cm"

CYRAD(1)=RDET;
CYRAD(2)=CYRAD(1)+RGAP;
CYRAD(3)=CYRAD(2)+RTCOV;

"PRINT OUT THE CYLINDER RADII"
OUTPUT ;(///,' CYLINDER RADII',/);
DO I=1,NCYL [
CYRAD2(I)=CYRAD(I)*CYRAD(I);
OUTPUT I,CYRAD(I);('  I=',I5,5X,' CYRAD(I)=',G15.5,'CM');
]
"*****************************************************************"
"********** STEP 5.  INITIALIZATION FOR AUSGAB COMES NEXT *********"
"*****************************************************************"

CALL ECNSV1(0,NREG,TOTKE);"  INITIALIZE ESUM ARRAY FOR ENERGY"
"                            CONSERVATION CALCULATION."
"                            NREG=NUMBER OF REGIONS"
"                            TOTKE=TOTAL KE (DUMMY VARIABLE HERE)"
"                                  (MUST BE REAL*8)"

CALL NTALLY(0,NREG);

NCOUNT=0;  "PARTICLE HISTORY COUNTER"
ILINES=0;  "INITIALIZE LINE-OUTPUT COUNTER"
DEPE=0.D0; "ZERO THE ENERGY DEPOSITION AT SCINTILATOR"
/PEF,TEF/=0.0;  "Zero the efficiency"

DO J=1,$NEBIN [PH(J)=0.0;]  "Zero the pulse-heght"
DO ND=1,$NDET [
DO J=1,$NEBIN [
/SPG(ND,J),SPE(ND,J),SPP(ND,J)/=0.0; "Zero the spectrum"
]]

"*****************************************************************"
"****** STEP 6.  DETERMINATION OF INCIDENT PARTICLE PROPERTIES ****"
"*****************************************************************"

IQI=0;  "INCIDENT PARTICLE"

EI=1.33 +ABS(IQI)*PRM;   "TOTAL ENERGY OF PARTICLE (MEV) "

AVAILE=EI + IQI*PRM;  "AVAILABLE K.E. (MEV) (MUST BE REAL*8)"
EKIN=AVAILE;
ECUTMN=ECUT(4); EK0=EKIN;  "*PRESTA*"
$PRESTA-INPUTS;  "INPUT THE *PRESTA* VARIABLES"

DELTAE=0.05;  "Energy bin of response"

XI=0.0; YI=0.0; ZI=0.0;  "STARTING COORDINATES (CM)"
UI=0.0; VI=0.0; WI=1.0;  "INCIDENT DIRECTION COSINES"
IRI=2;  "ENTRANCE REGION DEFINITION"
WTI=1.0;  "WEIGHT FACTOR OF UNITY"

IDINC=-1; "AN IDENTIFIER (LIKE IARG) TO MARK INCIDENT PARTICLES"

IXXST=17847465;
IXX=IXXST;  "INITIALIZED RANDOM NUMBER WITH STARTING SEED"

$RNG-INITIALIZATION;

NWRITE=10;  "NUMBER OF INCIDENT CASES TO PRINT OUT"

NCASES=$NCASES; "MAXIMUM NUMBER OF INCIDENT CASES TO RUN"
```

```
NBATCH=$NBATCH;  "NUMBER OF BATCH"
NCASPB=NCASES/NBATCH; "NUMBER OF CASES PER BATCH"
NOFBAT=0;  "NUMBER OF BATCH FINISHED"

NLINES=15; "NUMBER OF LINES TO PRINT OUT"


"*****************************************************************"
"******************** STEP 7. SHOWER-CALL---NEXT *****************"
"*****************************************************************"

DO NOFBAT=1,NBATCH [ "BATCH-LOOP"

DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
  OUTPUT EI,XI,YI,ZI,UI,VI,WI,
  IQI,IRI,IDINC;   (7G15.7,3I5);
  ILINES=ILINES+1;]

CALL SHOWER(IQI,EI,XI,YI,ZI,UI,VI,WI,IRI,WTI);

"If some energy is depositted inside detector add pulse-height"
"and efficiency"

IF(DEPE.GT.0.D0) [
IE=DEPE/DELTAE+1;
IF(IE.LE.$NEBIN) [PH(IE)=PH(IE)+WTI;]
IF(DEPE.GE.EI*0.999) [PEF=PEF+WTI;]
TEF=TEF+WTI;]
DEPE=0.D0;

NCOUNT=NCOUNT + 1;
IXXEND=IXX;   "LAST RANDOM NUMBER USED"

] "End of SHOWER CALL loop for each BATCH"

"Calcurate average value for this BATCH"

DO IE=1,$NEBIN [
PHPB(IE,NOFBAT)=PH(IE)/NCASPB;
PH(IE)=0.0;
]
PEFPB(NOFBAT)=PEF/NCASPB;
TEFPB(NOFBAT)=TEF/NCASPB;
/PEF,TEF/=0.0;

DO ND=1,$NDET [
DO IE=1,$NEBIN [
SPGPB(ND,IE,NOFBAT)=SPG(ND,IE)/NCASPB; "Gamma spectrum for this BATCH"
SPEPB(ND,IE,NOFBAT)=SPE(ND,IE)/NCASPB; "Electron spectrum for this BATCH"
SPPPB(ND,IE,NOFBAT)=SPP(ND,IE)/NCASPB; "Positron spectrum for this BATCH"
/SPG(ND,IE),SPE(ND,IE),SPP(ND,IE)/=0.0;
]]

]  "End of BATCH-loop"

TOTKE=NCOUNT*AVAILE; "TOTAL (AVAILABLE) K.E."

"*****************************************************************"
"******************** STEP 8. OUTPUT OF RESULTS ******************"
"*****************************************************************"

OUTPUT NCOUNT,IXXST,IXXEND,AVAILE,TOTKE;
('1',I10,' CASES COMPLETED',
     //,' IXXST=',I12,/,' IXXEND=',I12,//,' AVAILABLE K.E.=',
     G15.5,' MEV',/,' TOTKE=',E15.5,' MEV',//);

IF(IPLC.EQ.0.AND.IBCA.EQ.0) [
OUTPUT;(' PRESTA algorithm is used'/);]
ELSE [
OUTPUT;(' Default EGS4 calculation'/);]
```

```
OUTPUT TDE,RDET,TCOV,RTCOV,TGAP,RGAP;
(/' Detector length=',G15.5,' cm'/ ' Detector radius=',G15.5,' cm'/
' Al cover thickness=',G10.2,' cm'/ ' Al cover side thickness=',
G10.2,' cm'/' Front gap =',G10.2,' cm'/' Side gap =',G10.2,' cm'/);

OUTPUT EI;(' Results for ',G15.5,'MeV photon'/);

"Calculate average and its deviation"

/AVPE,DESCI2/=0.0;
DO J=1,NBATCH [
AVPE=AVPE+PEFPB(J)/NOFBAT;
DESCI2=DESCI2+PEFPB(J)*PEFPB(J)/NBATCH;
]
SIGPE=SQRT((DESCI2-AVPE*AVPE)/(NBATCH-1));
AVPE=AVPE*100.0;
SIGPE=SIGPE*100.0;
OUTPUT AVPE,SIGPE;(' Peak efficiency =',G15.5,'+-',G15.5,' %');

/AVTE,DESCI2/=0.0;
DO J=1,NBATCH [
AVTE=AVTE+TEFPB(J)/NOFBAT;
DESCI2=DESCI2+TEFPB(J)*TEFPB(J)/NBATCH;
]
SIGTE=SQRT((DESCI2-AVTE*AVTE)/(NBATCH-1));
AVTE=AVTE*100.0;
SIGTE=SIGTE*100.0;
OUTPUT AVTE,SIGTE;(' Total efficiency =',G15.5,'+-',G15.5,' %');


OUTPUT ;(/' Pulse height distribution ');
DO IE=1,$NEBIN [
ELOW=DELTAE*(IE-1);
EUP=DELTAE*IE;
IF(ELOW.GT.EKIN) [EXIT;]

/AVPH,DESCI2/=0.0;
DO J=1,NBATCH [
AVPH=AVPH+PHPB(IE,J)/NBATCH;
DESCI2=DESCI2+PHPB(IE,J)*PHPB(IE,J)/NBATCH;
]
SIGPH=SQRT((DESCI2-AVPH*AVPH)/(NBATCH-1));
AVPH=AVPH/DELTAE;
SIGPH=SIGPH/DELTAE;
OUTPUT EUP,AVPH,SIGPH;
(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
' counts/MeV/incident');
]
"Particle spectrum.  Incident particle spectrum to detector system"
"in this example."

OUTPUT;(/' Particle spectrum crossing the detector plane'/
30X,'particles/MeV/source photon'/
' Upper energy',11X,'  Gamma',18X,' Electron',14X,' Positron');

DO ND=1,$NDET [ "Spectrum at each detector plane"
DO IE=1,$NEBIN [
ELOW=DELTAE*(IE-1);
EUP=DELTAE*IE;
IF(ELOW.GT.EKIN) [EXIT;]

"Gamma spectrum per MeV per source"
/AVSPG,DESCI2/=0.0;
DO J=1,NBATCH [
AVSPG=AVSPG+SPGPB(ND,IE,J)/NBATCH;
DESCI2=DESCI2+SPGPB(ND,IE,J)*SPGPB(ND,IE,J)/NBATCH;
]
SIGSPH=SQRT((DESCI2-AVSPG*AVSPG)/(NBATCH-1));
AVSPG=AVSPG/DELTAE;
SIGSPH=SIGSPH/DELTAE;

"Electron spectrum per MeV per source"
/AVSPE,DESCI2/=0.0;
```

```
DO J=1,NBATCH [
AVSPE=AVSPE+SPEPB(ND,IE,J)/NBATCH;
DESCI2=DESCI2+SPEPB(ND,IE,J)*SPEPB(ND,IE,J)/NBATCH;
]
SIGSEL=SQRT((DESCI2-AVSPE*AVSPE)/(NBATCH-1));
AVSPE=AVSPE/DELTAE;
SIGSEL=SIGSEL/DELTAE;

"Positron spectrum per MeV per source"
/AVSPP,DESCI2/=0.0;
DO J=1,NBATCH [
AVSPP=AVSPP+SPPPB(ND,IE,J)/NBATCH;
DESCI2=DESCI2+SPPPB(ND,IE,J)*SPPPB(ND,IE,J)/NBATCH;
]
SIGSPO=SQRT((DESCI2-AVSPP*AVSPP)/(NBATCH-1));
AVSPP=AVSPP/DELTAE;
SIGSPO=SIGSPO/DELTAE;

OUTPUT EUP,AVSPG,SIGSPH,AVSPE,SIGSEL,AVSPP,SIGSPO;
(G10.5,' MeV--',3(G12.5,'+-',G12.5));
]
] "end of detector loop"

"NEXT, CALL THE SUBROUTINE ECNSV1 TO WRITE-OUT THE ENERGY DEPOSITION"
"TOTALS---TO CHECK ENERGY CONSERVATION FOR ONE THING"

CALL ECNSV1(1,NREG,TOTKE);

CALL NTALLY(1,NREG);

STOP;
END;  "END OF MAIN PROGRAM"



%E
"*********************************************************************"
"                                 STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE AUSGAB(IARG);
"                                 EGS4 SUBPROGRAM -  8 MAY 1983/1730"
"*********************************************************************"
;COMIN/DEBUG,EPCONT,ETALY1,LINES,MISC,NTALY1,PASSIT,STACK,TOTALS,USEFUL/;
REAL*8 DPWT,DEPE;
IRL=IR(NP);   "SET LOCAL VARIABLE"
DPWT=WT(NP);

"KEEP TRACK OF THE ENERGY DEPOSITION---FOR CONSERVATION PURPOSES"
ESUM(IQ(NP)+2,IRL,IARG+1)=ESUM(IQ(NP)+2,IRL,IARG+1)+EDEP*DPWT;
NSUM(IQ(NP)+2,IRL,IARG+1)=NSUM(IQ(NP)+2,IRL,IARG+1) + 1;

IF(MED(IRL).EQ.1) ["particle is inside the detector"
DEPE=DEPE+EDEP;  "Add energy deposition"
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
IF(IQ(NP).EQ.0) ["photon"
IE=E(NP)/DELTAE+1;
IF(IE.LE.$NEBIN) [SPG(1,IE)=SPG(1,IE)+DPWT;]]
ELSEIF(IQ(NP).EQ.-1) ["Electron"
IE=(E(NP)-RM)/DELTAE+1;
IF(IE.LE.$NEBIN) [SPE(1,IE)=SPE(1,IE)+DPWT;]]
ELSE ["Positron"
IE=(E(NP)-RM)/DELTAE+1;
IF(IE.LE.$NEBIN) [SPP(1,IE)=SPP(1,IE)+DPWT;]]
] "end of entering to detector"
] "end of inside detector"

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
OUTPUT E(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP),
  IQ(NP),IRL,IARG;   (7G15.7,3I5);
ILINES=ILINES+1;]

RETURN;
END;  "END OF SUBROUTINE AUSGAB"
```

```
%E
"*****************************************************************"
"                                STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE HOWFAR;
"                                EGS4 SUBPROGRAM -  8 MAY 1983/1730"
"*****************************************************************"
;COMIN/DEBUG,EPCONT,GEOM,PASSIT,STACK,THRESH/;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]

RETURN;
END;  "END OF SUBROUTINE HOWFAR"

%E
"*****************************************************************"
"                                STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE ECNSV1(NTREE,NREG,TOTKE);
"                                EGS4 SUBPROGRAM -  8 MAY 1983/1730"
"*****************************************************************"
"   SUBROUTINE FOR KEEPING TRACK OF ENERGY CONSERVATION---TO BE    "
"   USED WITH EGS USER CODES.  WHEN NTREE=0, THE PROGRAM IS        "
"   ENTERED IN ORDER TO INITIALIZE THE ESUM ARRAY TO ZERO.         "
"   OTHERWISE, IT IS ENTERED FOR TOTALING AND OUTPUTTING THE       "
"   RESULTS.  THE ESUM ARRAY IS NEEDED IN SUBROUTINE AUSGAB,       "
"   WHERE EDEP (ENERGY DEPOSITION) IS ADDED TO THE ELEMENT OF      "
"   THE ARRAY CORRESPONDING TO THE VALUE OF IQ, IR, AND IARG.      "
"*****************************************************************"
COMIN/DEBUG,ETALY1/;  "INSERT IN ALL SUBPROGRAMS THAT USE ESUM"
REAL*8 ROWSUM(4,$MXREG),COLSUM(4,5),SUMSUM(4),GSUM,TOTKE;

"   CHECK WHETHER NREG IS GE $MXREG.  IF IT IS, STOP AND OUTPUT.   "
IF(NREG.GT.$MXREG) [
MDUMMY=$MXREG;
OUTPUT NREG,MDUMMY;
(///,' ***** NOTE: STOPPED IN SUBROUTINE ECNSV1 BECAUSE NREG= ',
 I5,' IS LARGER THAN $MXREG= ',I5,' *****');
STOP;]

IF(NTREE.EQ.0) [  "INITIALIZE ESUM TO ZERO AND RETURN"
DO I=1,4 [DO J=1,NREG [DO K=1,5 [ESUM(I,J,K)=0.D0;]]]
RETURN;]

"   REACH THIS POINT WHEN FINAL TALLY IS TO BE MADE."

"   FIRST, INITIALIZE SUMS"

GSUM=0.D0;

DO IQ=1,4 [
SUMSUM(IQ)=0.D0;
DO IR=1,NREG [ROWSUM(IQ,IR)=0.D0;]
```

```
DO ICODE=1,5 [COLSUM(IQ,ICODE)=0.D0;]
"   END OF IQ-LOOP"]

"   SUM IQ=1,2,3 INTO IQ=4 OF ESUM FOR ALL IR- AND ICODE-VALUES"

DO IR=1,NREG [
DO ICODE=1,5 [
DO IQ=1,3 [
ESUM(4,IR,ICODE)=ESUM(4,IR,ICODE) + ESUM(IQ,IR,ICODE);
]]]

"   NORMALIZE DATA TO TOTKE"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ESUM(IQ,IR,ICODE)=ESUM(IQ,IR,ICODE)/TOTKE;
]]]

"   SUM-UP COLUMNS AND ROWS"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ROWSUM(IQ,IR)=ROWSUM(IQ,IR)+ESUM(IQ,IR,ICODE);
]]

DO ICODE=1,5 [
DO IR=1,NREG [
COLSUM(IQ,ICODE)=COLSUM(IQ,ICODE)+ESUM(IQ,IR,ICODE);
]]
"   END OF IQ-LOOP"]

"   NOW GET TOTAL FOR IQ AND GRAND TOTAL"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
SUMSUM(IQ)=SUMSUM(IQ)+ESUM(IQ,IR,ICODE);
IF(IQ.LE.3) [GSUM=GSUM+ESUM(IQ,IR,ICODE);]
]]]


"   NOW WRITE-OUT THE RESULTS OF THE ENERGY DEPOSITION SUMMARY"

DO IQ=1,4 [
IF(IQ.LE.3) [
IQNOW=IQ-2;
OUTPUT IQNOW;
('1ENERGY DEPOSITION SUMMARY FOR PARTICLES WITH IQ=',I2,///,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]
ELSE [
OUTPUT;  ('1ENERGY DEPOSITION SUMMARY FOR ALL PARTICLES:',///,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]

DO IR=1,NREG [
OUTPUT IR,(ESUM(IQ,IR,ICODE),ICODE=1,5),ROWSUM(IQ,IR);
  (I7,5X,5G15.7,5X,G15.7);
"   END OF IR-LOOP"]

OUTPUT (COLSUM(IQ,ICODE),ICODE=1,5),SUMSUM(IQ);
(/,3X,'COL SUM',2X,5G15.7,5X,G15.7);

"   END OF IQ-LOOP"]

OUTPUT GSUM; (/////,' TOTAL FRACTION=',G15.7,
'     NOTE: THIS NUMBER SHOULD BE VERY CLOSE TO UNITY');
```

```
RETURN;
END;  "END OF SUBROUTINE ECNSV1"

%E
"*****************************************************************"
"                                STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE NTALLY(NTREE,NREG);
"                                EGS4 SUBPROGRAM -  8 MAY 1983/1730"
"*****************************************************************"
"   THIS PROGRAM KEEPS 'TALLY' OF THE NUMBER OF TIMES AUSGAB IS    "
"   ENTERED FOR VARIOUS REASONS, ETC.  IT GIVES THE USER A 'ROUGH' "
"   IDEA OF WHETHER CERTAIN EVENTS ARE RARE OR NOT.                "
"   CAUTION *** DO NOT USE THESE NUMBERS IN ANY STATISTICAL SENSE] "
"*****************************************************************"
COMIN/DEBUG,NTALY1/;
INTEGER ROWSUM(4,$MXREG),COLSUM(4,5),SUMSUM(4),GSUM;

"   CHECK WHETHER NREG IS GE $MXREG.  IF IT IS, STOP AND OUTPUT."
IF(NREG.GT.$MXREG) [
MDUMMY=$MXREG;
OUTPUT NREG,MDUMMY;
(///,' ***** NOTE: STOPPED IN SUBROUTINE NTALLY BECAUSE NREG= ',
 I5,' IS LARGER THAN $MXREG= ',I5,' *****');
STOP;]

IF(NTREE.EQ.0) [  "INITIALIZE NSUM TO ZERO AND RETURN"
DO I=1,4 [DO J=1,NREG [DO K=1,5 [NSUM(I,J,K)=0;]]]
RETURN;]

"   REACH THIS POINT WHEN FINAL TALLY IS TO BE MADE."

"   FIRST, INITIALIZE SUMS"

GSUM=0;

DO IQ=1,4 [
SUMSUM(IQ)=0;
DO IR=1,NREG [ROWSUM(IQ,IR)=0;]
DO ICODE=1,5 [COLSUM(IQ,ICODE)=0;]
"   END OF IQ-LOOP"]

"   SUM IQ=1,2,3 INTO IQ=4 OF NSUM FOR ALL IR- AND ICODE-VALUES"

DO IR=1,NREG [
DO ICODE=1,5 [
DO IQ=1,3 [
NSUM(4,IR,ICODE)=NSUM(4,IR,ICODE) + NSUM(IQ,IR,ICODE);
]]]

"   SUM-UP COLUMNS AND ROWS"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ROWSUM(IQ,IR)=ROWSUM(IQ,IR)+NSUM(IQ,IR,ICODE);
]]

DO ICODE=1,5 [
DO IR=1,NREG [
COLSUM(IQ,ICODE)=COLSUM(IQ,ICODE)+NSUM(IQ,IR,ICODE);
]]
"   END OF IQ-LOOP"]

"   NOW GET TOTAL FOR IQ AND GRAND TOTAL"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
SUMSUM(IQ)=SUMSUM(IQ)+NSUM(IQ,IR,ICODE);
IF(IQ.LE.3) [GSUM=GSUM+NSUM(IQ,IR,ICODE);]
]]]
```

```
"   NOW WRITE-OUT THE RESULTS"

DO IQ=1,4 [
IF(IQ.LE.3) [
IQNOW=IQ-2;
OUTPUT IQNOW;
('1SUMMARY OF EVENT COUNT FOR PARTICLES WITH IQ=',I2,///,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]
ELSE [
OUTPUT;  ('1SUMMARY OF EVENT COUNT FOR ALL PARTICLES:',///,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]

DO IR=1,NREG [
OUTPUT IR,(NSUM(IQ,IR,ICODE),ICODE=1,5),ROWSUM(IQ,IR);
  (I7,5X,5I15,5X,I15);
"   END OF IR-LOOP"]

OUTPUT (COLSUM(IQ,ICODE),ICODE=1,5),SUMSUM(IQ);
(/,3X,'COL SUM',2X,5I15,5X,I15);

"   END OF IQ-LOOP"]

OUTPUT GSUM; (/////,' TOTAL NUMBER OF EVENTS=',I15);


RETURN;
END;  "END OF SUBROUTINE NTALLY"


%E
%C80
"-------------------------------------------------------------------"
" Start of PRSTAAUX MORTRAN - Auxiliary codes required by PRESTA. "
"--------------------------   Taken from NRCCAUXP MORTRAN.        "
"                            24 August 1989 WRN                   "
"-------------------------------------------------------------------"
"                                                                   "
"*******************************************************************"
"                            *       *                              "
"                            * FIXTMX *                              "
"                            *       *                              "
"                            *********                              "
SUBROUTINE FIXTMX(ESTEP,MEDIUM);
"                                                                   "
"  THIS ROUTINE CHANGES THE STEP SIZE ALGORITHM USED IN EGS SO THAT  "
"  THE STEP SIZE ARRAYS FOR TMXS CORRESPOND TO AN ARBITRARY,BUT      "
"  FIXED FRACTIONAL ENERGY LOSS ESTEPE.                             "
"  IT IS ONLY NECESSARY FOR LOW ENERGY ELECTRON PROBLEMS SINCE       "
"  TYPICALLY THE 200*TEFFO RESTRICTION ON TMXS IS MORE STRINGENT     "
"   FOR ELECTRONS WITH ENERGIES ABOVE A FEW MEV                     "
"                                                                   "
"  NOTE THAT THE $TMXS-OVER-RIDE MACRO MAY STILL BE IN FORCE IN EGS. "
"                                                                   "
"  THE ROUTINE CHANGES THE VALUES ONLY FOR THE MEDIUM 'MEDIUM'      "
"  AND IT SHOULD PROBABLY BE USED FOR ALL MEDIA IN A PROBLEM.       "
;"                                                                  "
"  THE ROUTINE MUST BE CALLED AFTER HATCH HAS BEEN CALLED AND BEFORE "
"  THE SIMULATION IS BEGUN.                                         "
"                                                                   "
"  THE ROUTINE IS INDEPENDENT OF WHAT UNITS ARE BEING USED, AS LONG  "
"  AS THEY ARE CONSISTENT( E.G. CM, RL OR G/CM**2 )                 "
"                                                                   "
"  IF CALLED WITH IOLDTM=0 (PASSSED IN COMIN USER) THE TMXS ARRAYS ARE"
"  ADJUSTED TO GIVE A FIXED ESTEPE AND ARE SUBJECTED TO THE TMIN AND "
"  CONSTRAINTS.                                                     "
"  IF CALLED WITH IOLDTM=1 THE CURRENT EGS ALGORITHM IS USED.       "
"  IF CALLED WITH IOLDTM=0 AND ESTEPE=0 THE CURRENT EGS ALGORITHM IS "
"  USED.                                                            "
"  IF CALLED WITH IOLDTM=1 AND ESTEPE=0 THEN ESTEPE=1.0 IS USED.    "
```

```
"                                                                    "
"  FOR A DETAILED DISCUSSION OF THE USE OF THIS ROUTINE, SEE         "
"  'Low Energy Electron Transport with EGS' in Nuclear Instr. and    "
"  Methods A227 (1984)535-548.  D.W.O. Rogers                        "
"                                                                    "
"  FOR A DISCUSSION OF THE NEW FEATURES (V03+) OF THIS ROUTINE,      "
"  ESPECIALLY WITH REGARD TO THE NEW UPPER AND LOWER LIMITS, SEE     "
"  'PRESTA-the Parameter Reduced Electron-Step Transport Algorithm-  "
"  for Electron Monte Carlo Transport' by A.F.Bielajew & D.W.O.Rogers,"
"  NRCC Internal Report PIRS-042 obtainable by contacting the above. "
"                                                                    "
"  V01        DEC 10,1981 DAVE ROGERS NRCC                          "
"  V02        DEC 1984                    EGS4 VERSION              "
"  V03        JAN 1986  ALEX BIELAJEW NRCC   REVISED FOR PRESTA     "
"********************************************************************"
;COMIN/ELECIN,MEDIA,USER/;

ESTEPE=ESTEP;

IF(MEDIUM > $MXMED)["ERROR"  OUTPUT MEDIUM;
(///'0********* MEDIUM=',I4,' IN FIXTMX IS TOO LARGE');RETURN;]

IF((ESTEPE = 0.) & (IOLDTM = 1)) RETURN; "USE THE CURRENT ALGORITHM "
IF(ESTEPE = 0.) ESTEPE=1.; "NEW VERSION DEFAULTS TO TOTAL ENERGY LOSS"
IF(IOLDTM = 0)[BLCCC=BLCC(MEDIUM);XCC2=XCC(MEDIUM)**2; "NEEDED BY ROOTMX"]
"SET UP SOME VARIABLES FOR FIRST PASS THROUGH LOOP"
EI =EXP( (1.-EKE0(MEDIUM))/EKE1(MEDIUM));"ENERGY OF FIRST TABLE ENTRY"
EIL = ALOG(EI);  LEIL=1;
"THIS IS EQUIVALENT TO $SETINTERVAL EIL,EKE; BUT AVOIDS ROUNDOFF"
$EVALUATE EDEDX USING EDEDX(EIL);"GET THE ELECTRON STOPPPING AT EI"
"NOW CALCULATE STEP REQUIRED TO CAUSE AN ESTEPE REDUCTION IN ENERGY"
IF(IOLDTM = 1)[SI=ESTEPE*EI/EDEDX;]ELSE[SI=ROOTMX(EI,ESTEPE);]
"TABULATED ENERGIES ARE IN A FIXED RATIO - CALC LOG OF THE RATIO"
ERATIO=-1./EKE1(MEDIUM);

NEKE=MEKE(MEDIUM);"NUMBER OF ELEMENTS IN STORAGE ARRAY"
DO I=1,NEKE-1[
EIP1=EXP((FLOAT(I+1)-EKE0(MEDIUM))/EKE1(MEDIUM));"ENERGY AT I+1"
EIP1L=ALOG(EIP1);LEIP1L=I+1;"DESIGNED THIS WAY=$SETINTERVAL"
$EVALUATE EDEDX USING EDEDX(EIP1L);
IF(IOLDTM = 1)[SIP1=ESTEPE*EIP1/EDEDX;]ELSE[SIP1=ROOTMX(EIP1,ESTEPE);]


"NOW SOLVE THESE EQUATIONS                                           "
"  SI   = TMXS1 * EIL   + TMXS0                                      "
"  SIP1 = TMXS1 * EIP1L + TMXS0                                      "
"                                                                    "
TMXS1(I,MEDIUM)=(SI-SIP1)/ERATIO;TMXS0(I,MEDIUM)=SI-TMXS1(I,MEDIUM)*EIL;
"TRANSFER VALUES FOR NEXT LOOP"
EIL=EIP1L;SI=SIP1;]
"NOW PICK UP LAST TABLE ENTRY WHICH APPLIES ONLY TO LAST ENERGY"
TMXS0(NEKE,MEDIUM)=TMXS0(NEKE-1,MEDIUM);
TMXS1(NEKE,MEDIUM)=TMXS1(NEKE-1,MEDIUM);
RETURN;END;
%E


"********************************************************************************"
"                                 *        *                                     "
"                                 * ROOTMX *                                     "
"                                 *        *                                     "
"                                 **********                                     "
"                                                                                "
FUNCTION ROOTMX(EI,ESTEP);
"                                                                                "
"        THIS ROUTINE RETURNS MAX(TMIN,MIN(TMAX,ESTEPE*EI/DEDX)) WHERE           "
"    TMAX IS THE MAXIMUM STEP ALLOWED BY THE MOLIERE MULTIPLE SCATTERING         "
"    THEORY, TMIN IS THE THE MINIMUN STEP AND ESTEPE*EI/DEDX IS THE GREATEST     "
"    STEP ALLOWED DUE TO CONTINUOUS ENERGY LOSS PROCESSES.                       "
"                                                                                "
"        NOTE THE USE OF ITS AUXILLIARY FUNCTION FTMX APPENDED TO ROOTMX.        "
"    BECAUSE THE TMAX FUNCTION IS STRONGLY ENERGY DEPENDENT, IT WAS FOUND        "
"    NECESSARY TO INCLUDE A CORRECTION FOR ENERGY LOSS IN IT. OTHERWISE THE      "
```

```
"    UPPER LIMIT COULD BE GREATLY EXCEEDED - BY AS MUCH AS 50% IN SOME CASES.  "
"    CORRECTING FOR ENERGY LOSS NECESSITATES USING A ROOT FINDING METHOD TO    "
"    OBTAIN TMAX (HENCE THE NAME ROOTMX). TMIN IS ALSO STRONGLY ENERGY         "
"    DEPENDENT BUT IT DOES NOT MATTER WITHIN THE LOGIC OF THE CODE IF THIS     "
"    QUANTITY IS AS MUCH AS 50% HIGH SINCE NO PHYSICS CONSTRAINTS WILL BE      "
"    VIOLATED.                                                                 "
"                                                                              "
"        THE ZERO-FINDING ROUTINE IS A CRUDE ONE BASED ON THE ASSUMPTION THAT  "
"    THE FUNCTION FTMX IS MONOTONIC AND THAT THE FUNCTION EVALUATED AT THE TWO "
"    STARTING POINTS RETURNS DIFFERENT SIGNS. IF THE SIGNS ARE THE SAME THEN   "
"    EITHER THE ENERGY-LOSS STEP-SIZE IS MORE RESTRICTIVE OR THE STEP-SIZE IS  "
"    BELOW TMIN.                                                               "
"                                                                              "
"        ALTHOUGH THIS ROUTINE COMES WITH THE PRESTA PACKAGE IT IS REALLY      "
"    INDEPENDENT OF IT AND IT IS AN IMPROVEMENT OVER THE PREVIOUS TMXS METHODS."
"    THE OLD TMXS ROUTINE ALLOWED BOTH THE TMAX AND TMIN BOUNDS TO BE VIOLATED."
"    EXCEEDING TMAX TAKES ONE OUT OF THE REGION OF VALIDITY OF THE MOLIERE     "
"    THEORY STILL ALLOWING A MULTIPLE SCATTERING SELECTION BUT OF UNPREDICTABLE"
"    WORTH. GOING LOWER THAN TMIN CAUSES THE MULTIPLE SCATTERING TO GET        "
"    SWITCHED OFF (STARTING WITH THE LOWER ENERGIES). THIS CAN SOMETIMES LEAD  "
"    TO CALCULATIONAL ARTIFACTS. ONE WORD OF CAUTION] USING THIS ROUTINE AT    "
"    VERY LOW ELECTRON ENERGIES .LE.10 keV CAUSES NEGATIVE USTEP ERRORS IF THE "
"    OLD EGS PATHLENGTH CORRECTION ALGORITHM (BASED ON FERMI-EYGES THEORY) IS  "
"    USED. THE OLD EGS LESSENED THIS PROBLEM BY REDUCING THE UPPER LIMIT TO    "
"    0.8 THE VALUE USED IN THIS ROUTINE. THE PRESTA PATHLENGTH CORRECTION DOES "
"    NOT GIVE NEGATIVE USTEPS IN ANY OF THE CASES WE HAVE TESTED.              "
"                                                                              "
"             VERSION 1       ALEX BIELAJEW           JAN. 86                  "
"             VERSION 1.1     ALEX BIELAJEW           OCT. 87                  "
"                             Lower limit ESTEPE violation fixed               "
"                                                                              "
"******************************************************************************"

;
COMIN/USEFUL,USER/;
ESTEPE=ESTEP;
TMIN=2.718282*EI*(EI+2.*RM)/(BLCCC*(EI+RM)**2); "LOWER LIMIT, eq.(2-8)"
X1=TMIN; "INITIAL LOWER STARTING POINT OF THE SEARCH"
X2=ESTEPE*EI/EDEDX; "INITIAL UPPER STARTING POINT OF THE SEARCH"

"THIS IS THE FIX-UP FOR THE MINIMUM STEP-SIZE"
IF( X2 <= X1 ) [ROOTMX=X1;RETURN;]

F1=FTMX(X1,EI);F2=FTMX(X2,EI);
AF1=ABS(F1);AF2=ABS(F2);
SF1=SIGN(1.,F1);SF2=SIGN(1.,F2);

"FIRST CHECK TO SEE IF EITHER OF THE STARTING POINTS IS ALREADY GOOD ENOUGH."
 IF((AF1 <= $ROOTMX_PRECISION) | (AF2 <= $ROOTMX_PRECISION))[
    IF(AF1 <= AF2)[ROOTMX=X1;]ELSE[ROOTMX=X2;]]

"NOW CHECK TO SEE IF EITHER THE ENERGY LOSS IS MORE RESTRICTIVE THAN THE       "
"UPPER LIMIT TMAX (TRUE FOR HIGH ENERGIES) OR IF IT MORE RESTRICTIVE THAN     "
"TMIN (TRUE FOR LOW ENERGIES WITH A SMALL ENOUGH ESTEPE).                     "
ELSEIF(SF1 = SF2)[ROOTMX=X2;]

"OTHERWISE A SEARCH FOR TMAX MUST BE UNDERTAKEN."
ELSE[ "ITERATE"
ITI=0; "NUMBER OF ITERATIONS COUNTER"
XL=X1; "LAST X FOUND"
:SEARCH-ROOT:LOOP[
ITI=ITI+1;
IF(ITI > 1000)[ "QUIT IF THIS HAPPENS"
OUTPUT;(' SEARCH FOR TMXS ABORTED. TOO MANY ITERATIONS');STOP;]
XT=(X1*F2-X2*F1)/(F2-F1);
IF(XT = XL)[ROOTMX=XT;EXIT:SEARCH-ROOT:;"CONVERGENCE OBTAINED"]
FT=FTMX(XT,EI);AFT=ABS(FT);
IF(AFT <= $ROOTMX_PRECISION)[ROOTMX=XT;EXIT:SEARCH-ROOT:;"CONVERGENCE OBTAINED"]
ELSE[ "RE-ITERATE"
SFT=SIGN(1.,FT);
IF(SFT = SF1)[X1=XT;F1=FT;AF1=AFT;SF1=SFT;]ELSE[X2=XT;F2=FT;AF2=AFT;SF2=SFT;]
XL=XT; "UPDATE LAST X FOUND"
]
```

```
] "END OF SEARCH FOR ROOT LOOP"
] "END OF ITERATE ELSE"
RETURN;END;

FUNCTION FTMX(T,EI);
"When t=tmax as defined in eq.(2-10) this function returns 0. It is used by    "
"FUNCTION ROOTMX in the search for tmax.                                        "
COMIN/USEFUL,USER/;
"Energy dependent quantities are evaluated at the energy mid-point of the step."
"See section IV of the report PIRS-042.                                        "
EK=AMAX1(0.0001,EI-0.5*EDEDX*T);E=EK+RM;BETA2=EK*(E+RM)/E**2;
A=BLCCC/BETA2;G=XCC2/(E*BETA2)**2;
FTMX=1./ALOG(A/G)-G*T;
RETURN;END;
%E

"*****************************************************************"
SUBROUTINE RMARIN;
"*****************************************************************"
COMIN/RANDOM/;

IF((IXX.LE.0).OR.(IXX.GT.31328)) IXX=1802; "SETS MARSAGLIA DEFAULT"

" BUG. In the following line the assignment previous to 90/09/18  "
" was to IXX. This DID NOT upset the randomness of the sequence,  "
" just the initial starting point. BLIF 90/09/18.                 "

IF((JXX.LE.0).OR.(JXX.GT.30081)) JXX=9373; "SETS MARSAGLIA DEFAULT"

I = MOD(IXX/177,177) + 2;
J = MOD(IXX,    177) + 2;
K = MOD(JXX/169,178) + 1;
L = MOD(JXX,    169)    ;

DO II=1,97[
    S=0.0;T=0.5;
    DO JJ=1,24[
        M=MOD(MOD(I*J,179)*K,179);
        I=J;J=K;K=M;L=MOD(53*L+1,169);
        IF(MOD(L*M,64).GE.32) S=S+T;
        T=0.5*T;
        ]
    URNDM(II)=S;
    ]

CRNDM  =   362436./16777216.;
CDRNDM =  7654321./16777216.;
CMRNDM = 16777213./16777216.;

IXX = 97;
JXX = 33;

RETURN;END;

"*****************************************************************"
"************************ UCNAI3P END ****************************"
"*****************************************************************"
%N
```