KEK Internal 99-4
June 1999
H/R/D

Lecture Note

Installation of EGS4

(Revised in 1999)

(Last modified in 04 OCT 2002)

Y. Namito, H. Nakamura and H. Hirayama

High Energy Accelerator Research Organization (KEK)

Lecture Note

Installation of EGS4

(Revised in 1999)

Y. Namito, H.Nakamura and H. Hirayama
High Energy Accelerator Research Organization (KEK),
Oho, Tsukuba-shi, Ibaraki-ken, 305, Japan

Abstract

`egs4unix_kek` is distributed by KEK. This package is made of selected files from NRCC's `egs4unix_2.0` and expansions developed at KEK plus simplified C shell scripts. `egs4pc_kek` and `egs4macintosh_kek` are also distributed by KEK. These packages are also made of selected files from NRCC's `egs4unix_2.0` plus Batch files for PC or shell script for Macintosh. In this report, the way to install EGS4 to a unix workstation (SUN, HP, DEC-ULTRIX, Silicon-Graphics, IBM, Hitachi and Linux), PC(DOS/V and 98: Japanese previous standard) and Macintosh using `egs4unix_kek`, `egs4pc_kek` and `egs4macintosh_kek` is described. The EGS4 system consists of MORTRAN, EGS4, PEGS4 and the user's code. The goal of installing EGS4 is to run a user's code named `ucsampl4.mor` and to obtain the same output as that shown in SLAC-265. The goal of installing PEGS4 is to obtain an output from `examin.mor` which is equivalent to that shown in this report. In the case of a PC, a g77, Lahey LF90, Microsoft Fortran/Power station or Visual Fortran is supposed to be used as a compiler. In the case of a Macintosh, an LS-Fortran and an MPW shell are supposed to be used as a compiler and shell language. This report may be useful even if the users use a different compiler other than those described in this report, since several suggestions concerning that case are described. This report describes the equivalent contents in both Japanese and English.

Contents

English Part

1 Installation to a Unix computer

1.1 Get the system

From the FTP server of KEK, get egs4unix_kek from an anonymous ftp. Connect to Node name: ftp.kek.jp (130.87.34.29) by ftp and enter "anonymous" as the user's ID and "Your ID@node name" as the password to login. Then, change the mode to binary by

```
ftp> binary
```

and obtain the following files by the get command:

/kek/kek_egs4/egs4unix_kek/egs4unix_kek.tar.Z	(EGS4 system)
/kek/kek_egs4/egs4unix_kek/user_code.tar.Z	(EGS4 sample user's code)

1.2 Extraction of a file

The obtained files are tar files compressed by the compress command. Move the files to the directory which you use as an EGS4 directory.

```
% uncompress egs4unix_kek.tar.Z
% uncompress egs4user_code.tar.Z
```

will uncompress the files and remove ".Z" from the file names. Then, extract each file,

```
% tar xvf egs4unix_kek.tar
% tar xvf user_code.tar
```

Afterwards, the input is shown in a line with a "%" prompt at the beginning.

1.3 Installation of MORTRAN

The MORTRAN system consists of mortran3.f and mornew77.raw; mornew77.raw is a data file of rules to convert a Mortran program to a Fortran program. By compiling mortran3.f, you obtain an executable file mortran3; mortran3 runs in following two modes:

Mode I: Generation of mortran3.dat mode

This mode is utilized in Mortran installation; mornew77.raw is converted in Hexagonal expression, and the converted data file is saved as mortran3.dat.

Mode II: Mortran/Fortran conversion mode

This mode is utilized in the usual EGS4 calculation. By using mortran3.dat, mortran3 converts a file written in Mortran language into a file written in Fortran language.

1.3.1 Generation of mortran3.dat (Mode I, machine dependence)

Compile `mortran3.f` without any change, since `mortran3.f` is set as MODE-I when it is distributed,

```
 % f77 -o mortran3 -O mortran3.f                   (For SUN)
 % f77 -o mortran3 -OK mortran3.f                  (For HP)
 % f77 -o mortran3 -static -O mortran3.f           (For DEC ULTRIX)
 % f77 -o mortran3 -static mortran3.f              (For Silicon Graphics)
 % f77 -o mortran3 mortran3.f                      (For IBM AIX)
 % f77 -o mortran3 mortran3.f                      (For Hitachi's HI-UX).
 % g77 -o mortran3 -O -fno-automatic mortran3.f    (For Linux)
```

Please be careful concerning different compiler options for each OS. The purposes of the options are:

- Automatic save of local variables. ("`-K`" for HP, "`-static`" for DEC ULTRIX and Silicon Graphics and "`-fno-automatic`" for g77 in Linux.) In the case of SUN and Hitachi, this is the default.
- Optimize. "`O`"(Alphabet) specifies optimization. In the case of a fortran compiler on Hitachi's HI-UX, some level of optimize is set as the default. You must increase the optimize level step by step while checking whether the calculation result has been changed.

In the case that you use a fortran compiler other than those listed above, specify the compiler option for the equivalent purpose.

After compiling, you obtain an executable file named `mortran3`. By keying in,

```
% mortran3
```

this `mortran3` is executed to produce `mortran3.dat`.

The first line of the obtained `mortran3.dat`,

```
v....2C USER F77 11JUN85
```

should be erased. You should erase the whole line. Open the `vi` editor,

```
% vi mortran3.dat
```

and type

```
:1d
```

The first line is thus removed. (BE CARFULL NOT TO LEAVE A BLANK LINE AT THE TOP OF THE FILE!) Then write the file and quit the editor by typing

```
:wq
```

You then come back to the Unix prompt.

If you did not obtain mortran3.dat, open mortran3.f by the vi editor to check if

```
IMODE=1
```

is written. If you find a line of CALL IOINIT, comment out that line, since it does not work on machines other than a SUN.

Once mortran3.dat is created in the installation, mortran3 is never used in Mode-I afterwards. mortran3 is changed into Mode-II for the usual egs4 calculation. Open mortran3.f using the editor, and change the line IMODE=1 to

```
IMODE=2
```

Then compile again.

```
 % f77 -o mortran3 -O mortran3.f                 (For SUN)
 % f77 -o mortran3 -OK mortran3.f                (For HP)
 % f77 -o mortran3 -static -O mortran3.f         (For DEC ULTRIX)
 % f77 -o mortran3 -static mortran3.f            (For Silicon Graphics)
 % f77 -o mortran3 mortran3.f                    (For IBM AIX)
 % f77 -o mortran3 mortran3.f                    (For Hitachi's HI-UX)
 % g77 -o mortran3 -O -fno-automatic mortran3.f  (For Linux)
```

The newly obtained mortran3 runs in Mode-II.

1.3.2 Modification of the compiler option (machine dependence)

Please skip this part when Hitachi's HI-UX is used as the default compiler, and the option has already been set for it.

When you use workstation with an OS other than Hitachi's HI-UX or IBM AIX, change the line of f77 mortjob.f in mortrun and egs4run in the same way as for compile in the previous section,

```
% f77  -O                 mortjob.f    (For SUN)
% f77  -OK                mortjob.f    (For HP)
% f77  -static -O         mortjob.f    (For DEC ULTRIX)
% f77  -static            mortjob.f    (For Silicon Graphics)
% g77  -O -fno-automatic  mortjob.f    (For Linux)
```

The auto-double option is equipped on Hitachi's or IBM's unix WS. You can use them when necessary.

In the case that you use a compiler other than those listed above, you must find and specify the equivalent options. Only two machine dependencies are a modification of the compiler option described in the previous section and that in this section. All of the procedures afterwards are machine independent.

1.3.3 Check of mortran3 (Mode 2) using check77.mor

A short program (10 lines) named `check77.mor` is used to check normal operation of the Mortran system. If `check77.mor` is converted into a Fortran program normally and compiled and executed without any problem, you can judge that the Mortran system is set up correctly. To execute `check77.mor`, type in

```
% mortrun check77.mor
```

Be CARFUL concerning the sixth letter of `mortrun`. It is not a but u! This `mortrun` is a C-shell script used to convert a program written in Mortran language into a program written in Fortran, and to compile and execute.

When the job is terminated, you will see

```
  **** Results of check77.f ***
   THE FIRST TEN INTEGERS AND THEIR SQUARES:

   I=  1 ( ODD INTEGER)    1
   I=  2 (EVEN INTEGER)    4
   I=  3 ( ODD INTEGER)    9
   I=  4 (EVEN INTEGER)   16
   I=  5 ( ODD INTEGER)   25
   I=  6 (EVEN INTEGER)   36
   I=  7 ( ODD INTEGER)   49
   I=  8 (EVEN INTEGER)   64
   I=  9 ( ODD INTEGER)   81
   I= 10 (EVEN INTEGER)  100
```

displayed. If you do not obtain this output, you must check the `mortjob.f` and `mortjob.list` using the editor. Check whether all of the macro and mortran statements have been converted into Fortran statement.

1.4 Installation of EGS4

No modification is necessary for egs4 files when installing egs4. You must just check that egs4 files (ex. `egs4.mor`) run without any trouble.

1.4.1 Running egs4run

You run a sample user's code, `ucsampl4.mor`, using a C-shell script, `egs4run`, to check whether egs4 runs without any trouble. As material data, `sampl4.dat` is used.

Execute `egs4run`,

```
% egs4run ucsampl4.mor sampl4.dat
Do you want to include MACRO in the mortran list?  yes
Do you use PRESTA? no
```

If you do not specify the user's code and material data, `egs4run` asks you for them. In the case that you can not execute `egs4run`, check its execution property. The content of `egs4run` is shown in Fig. 1.

When the dialog part of egs4run is finished, the following lines are displayed:

```
mortran3 procedure entered
f77 procedure entered
a.out procedure entered
egs4run procedure finished
```

and a prompt appears. Type in

```
% ps
```

to see the execution condition a.out. While it is being executed, the executing file and CPU time consumed for the process are indicated,

```
PID   TT  STAT  TIME COMMAND
***   **  ***   *:**  a.out
```

If you use g77, please ignore following warning message.

```
mortjob.for:  In block-data ‘_BLOCK_DATA__’:  mortjob.for:468:  warning:
COMMON/ELECIN/ EKELIM, EKE0(10),EKE1(10),CMFP0(10),CMFP1(10),RANGE_
Înitialization of large (108772-unit) aggregate area ‘elecin’ at ()̂
currently very slow and takes lots of memory during g77 compile -- to be
improved in 0.6
```

When the execution of a.out is finished, compare the calculation result in mortjob.output6 [1] with the corresponding output described in SLAC-265 (shown as Fig. 4). If they are the same, you can judge whether EGS4 is correctly installed. Run egs4run similarly when you make your own user's code and material data.

If the output is not the same as in Fig. 4, or the output is stopped in the middle, or you get an error message, some problem has occurred. You must grasp what has happened and fix the reason for the error. To do this, investigate other output files (see section 4.2 for the contents) and execute the content of egs4run one by one according to the next part.

1.4.2 Execution by individual commands

Skip this part in the case ucsampl4.mor runs without any trouble. Do this part only when you have some trouble executing egs4run, or you want to understand the contents of egs4run.

This part describes how to run user's code ucsampl4.mor using an individual Unix command. The execution of EGS4 is divided into three parts:

1. Catinate the user's code and EGS4 files into one file.

2. Convert a Mortran file into a Fortran file.

[1] You can change output into output6 or out by changing the open statement for unit 6 in ucsampl4.mor.

3. Compile and execute a Fortran file.

First check if the following `open` statements are contained in a macro in STEP 1 of `ucsampl4.mor`. If some lines are missing, add them.

```
open(unit=6 ,file='mortjob.output');
open(unit=8 ,file='mortjob.dummy');
open(unit=12,file='mortjob.xsec');
```

Then check if a line of `$OPEN;` is written in STEP.2. (This line is a template to be replaced by OPEN statements written in STEP 1.)

Then catinate the user's code and files of `egs4` into one big mortran file named `mortjob.`
`mortran`. Also, copy the material data to a file named `mortjob.xsec`.

```
% cat egs4mac.mortran  kek4mac.mortran  ucsampl4.mor\
% egs4blok.mortran egs4.mortran > mortjob.mortran
% cp sampl4.dat mortjob.xsec
```

Here, the "\ " character indicates continuing of a command line. Some terminal or printer indicates this using the "¥" character.

By typing

```
% mortran3
```

`mortran3` runs. It reads in `mortjob.mortran` and outputs `mortjob.f`. Compile and execute this fortran file,

```
% f77 mortjob.f
% a.out
```

You should then obtain the same output, in `mortjob.output` as shown in Fig. 4.

1.5 Installation of PEGS4

1.5.1 Generation of a PEGS4 executable file

Generate and keep an executable file of PEGS4 using `pegs4compile`. Key in

```
% pegs4compile
```

to compile `pegs4.mortran` and keep the executable file as `pegs4.exe`. The content `pegs4compile` is shown in Fig. 5. Usually, PEGS4 is not modified afterwards.

If you experience any trouble during the compile, open `pegs4.mortran` by editor to check that `CALL IOINIT` is commented out. At that time also check that the following OPEN statement is contained:

```
OPEN(5,  file='pgs4job.pegs4inp');
OPEN(6,  file='pgs4job.pegs4lst');
OPEN(7,  file='pgs4job.pegs4dat');
OPEN(8,  file='pgs4pepr.dat');
OPEN(9,  file='pgs4form.dat');
OPEN(10, file='pgs4job.pegs4err');
OPEN(21, file='pgs4job.pegs4plot');
OPEN(22, file='aprime.data');
```

1.5.2 Test execution of PEGS4

Using `pegs4run`, execute `pegs4.exe` to produce material data as a test run. Here, `sampl4.inp` is used as sample input data. You must make input data when you run PEGS4 afterwards.

```
% pegs4run sampl4.inp pegs4.output
```

Here, the first and the second operands specify the input file name and output file name, respectively. The content of `sampl4.inp` is shown in Fig. 9.

If execution of `pegs4run` is denied, check its execution property by

```
% ls -l pegs4run
```

1.5.3 Use of EXAMIN.MOR to check the PEGS4 output

PEGS4 automatically sets energy bins and linearly interpolates physical quantities (ex. total cross section, branching ratio et al), and then outputs the coefficients . It is necessary to compare the PEGS4 outputs on a new computer (to install PEGS4) and a standard computer. However, it is sometimes difficult to compare PEGS4 outputs directly. Due to some reasons, such as a difference in the precision, different energy bins may be used on a new computer and a standard one. In that case, identical physical quantities are described quite differently. For this comparison, an EGS4 user's code, `examin.mor`, is used to check the PEGS4 output. `examin.mor` reads in PEGS4 data and output them in a tabular form so that man can easily read then terminate without calling a shower. To run `examin.mor`, type in

```
% egs4run examin.mor pegs4.output
```

Here, `PRESTA` is not used.

Compare the newly generated `mortjob.output6` and `examin.out`,

```
% diff examin.out mortjob.output6
```

If they are the same or differ in the 5th digits several times, PEGS4 on the new computer runs in the same way as PEGS4 on the standard computer. If they differ more apparently, you must investigate the reason. `examin.out` is an output file from `examin.mortran`. Here, the input of `examin.mor` is calculated using PEGS4 with `sampl4.inp` (=input file) and pgs4pepr.dat[2] (= Stom and Israel's photoeffect cross section data) on KEK's central computer(ccce5=Hitach's WS). The content of examin.out is shown in Figs. 10 and 11.

1.6 egs4_unix

egs4_unix ver.3.0 is distributed from NRCC (Canada) and LBL. It consists of EGS4 (originally developed at SLAC), NRCC's improvements (ex. PRESTA: Improvement on low energy electron transport) and user's codes developed at NRCC. All of them were tested or modified to run on many unix computers (SUN、IRIS、HP、DEC、IBM R-series、Linux).

egs4_unix ver.3.0 is very convenient regarding its automatic installation feature. To avoid using OPEN statements in the user's code, a machine-dependent command is used in the C shell scripts in egs4_unix ver.3.0. It is thus necessary to change the C shell scripts for different computers. When the compiler or OS is changed, the C shell scripts must also to be changed. However, it is not easy to change the C shell scripts, since they are relatively long (a few hundred lines.). Although the NRCC/LBL distribution is convenient at its automatic installation, once some problem occurs, it is sometimes hard to fix the problem for a unix beginner.

On the other hand, only one machine-dependent part of the KEK distribution system (egs4unix_kek) is a specification of the compiler options. While the KEK distribution system always asks users to do a some jobs during installation based on basic unix knowledge, it is easy to fix a problem when the OS or compiler is changed.

egs4unix ver.3.0 also includes an executable file of MORTRAN and PEGS4 and C shell scripts to compile and execute egs4 (egs4.compile and egs4.run). To get egs4unix ver 3.0, access the EGS4 home page (http://ehssun.lbl.gov/egs/egs.html).

[2] PEGS4 uses photoeffect and pair-production cross sections based on Storm & Israel's compilation value (pgs4pepr.dat). This pgs4pepr.dat is included in egs4unxi_kek, egs4pc_kek and egs4macintosh_kek. On the other hand, Sakamoto of JAERI provided pgs4phtx.dat, which is a photoeffect and pair-production data file based on new photon cross-section data PHOTX (See "Proceedings of the Third EGS4 User's Meeting in Japan", KEK Proceedings 93-15 (1993) 77-82.{In Japanese}). This pgs4phtx.dat is also included in egs4unxi_kek, egs4pc_kek, and egs4macintosh_kek. To use pgs4phtx.dat instead of pgs4pepr.dat, change the files name in the open(unit=8) statement in pegs4.mor from pgs4pepr.dat to pgs4phtx.dat.

2 Installation to PC

2.1 Get the system

From the FTP server of KEK get egs4pc_kek by an anonymous ftp. Connect to Node name: ftp.kek.jp (130.87.34.29) by FTP and enter "anonymous" as the user's ID and "Your ID@node name" as the password to login. Then change the mode to binary by

```
ftp> binary
```

and get the following files using the get command:

```
/kek/kek_egs4/egs4pc_kek/egs4pc.exe     (EGS4 system)
/kek/kek_egs4/egs4pc_kek/user_cod.exe   (EGS4 sample user's code)
```

2.2 Extraction of a file

The obtained files are self-extraction files made using LHA. Move these files to the directory which you use as an EGS4 directory. (For example, C:¥EGS4LECT.)

```
C:¥EGS4LECT> EGS4PC.EXE
C:¥EGS4LECT> USER_COD.EXE
```

will extract individual files.

Afterwards, the input is indicated by a line with "C:¥EGS4LECT" prompt at the beginning.

2.3 Installation of MORTRAN

The MORTRAN system consists of MORTRAN3.FOR and MORNEW77.RAW. MORNEW77.RAW is a data file of rules to convert a Mortran program to a Fortran program. By compiling MORTRAN3.FOR, you get executable file MORTRAN3.EXE. MORTRAN3.EXE runs in the following two modes:

Mode I: Generation of MORTRAN3.DAT mode

> This mode is utilized during Mortran installation. MORNEW77.RAW is converted in Hexagonal expression, and the converted data file is saved as MORTRAN3.DAT.

Mode II: Mortran/Fortran conversion mode

> This mode is utilized in the usual EGS4 calculation. Using MORTRAN3.DAT, MORTRAN3.EXE convert a file written in Mortran language into a file written in Fortran language.

2.3.1 Generation of MORTRAN3.DAT (Mode I, machine dependence)

Compile MORTRAN3.FOR without any change, since MORTRAN3.FOR is set as MODE-I when it is distributed, [3]

```
C:¥EGS4LECT> LF90 MORTRAN3.FOR                       (For Lahey Fortran)
C:¥EGS4LECT> FL32 MORTRAN3.FOR                       (For MS-Fortran/PS)
C:¥EGS4LECT> F90 MORTRAN3.FOR                        (For Visual Fortran Ver.5)
C:¥EGS4LECT> DF MORTRAN3.FOR                         (For Visual Fortran Ver.6.1)
C:¥EGS4LECT> g77 -o mortran3 -O -fno-automatic  -fno-backslash mortran3.for
                                                     (For g77 on windows)
```

After compiling, you obtain an executable file named MORTARAN3.EXE. By typing

```
C:¥EGS4LECT> MORTRAN3
```

MORTRAN3.EXE is executed to produce MORTRAN3.DAT.

The first line of the obtained MORTRAN3.DAT,

```
v....2C USER F77 11JUN85
```

should be erased using the editor (MIFES, WZ etc.).
(BE CARFULL NOT TO LEAVE A BLANK LINE AT THE TOP OF THE FILE!)

If you do not obtain MORTRAN3.DAT, open MORTRAN3.FOR using an editor to check if

```
IMODE=1
```

is written. If you find a line of CALL IOINIT, comment out that line, since it does not work on machines other than SUN.

Once MORTRAN3.DAT is created in the installation, MORTRAN3.EXE is never used in Mode-I afterwards. MORTRAN3.EXE is changed into Mode-II for the usual egs4 calculation. Open MORTRAN3.FOR by an editor, and change the line of IMODE=1 to

```
IMODE=2
```

Then compile again.

[3] Before this, FORTRAN compiler must be installed and any necessary initialization must be performed. For example, if one installed g77 for Windows (http://www.geocities.com/Athens/Olympus/5564) to C:¥g77 directory, following command must be executed in DOS-window for initialization;

```
C:¥EGS4LECT> c:¥g77¥g77setup
```

See compiler manuals for details.

```
C:¥EGS4LECT> LF90 MORTRAN3.FOR                      (For Lahey Fortran)
C:¥EGS4LECT> FL32 MORTRAN3.FOR                      (For MS-Fortran/PS)
C:¥EGS4LECT> F90 MORTRAN3.FOR                       (For Visual Fortran Ver.5)
C:¥EGS4LECT> DF MORTRAN3.FOR                        (For Visual Fortran Ver.6.1)
C:¥EGS4LECT> g77 -o mortran3 -O -fno-automatic  -fno-backslash mortran3.for
                                                    (For g77 on windows)
```

The newly obtained `mortran3` runs in Mode-II.

2.3.2 Modification of the compiler name and option (machine dependence)

Skip this section if you use g77.

`MORTRUN.BAT`, `EGS4RUN.BAT`, `EGS4RUNP.BAT` and `PEGS4COM.BAT` are set for g77 as a default. If you use any other fortran compiler, comment out lines for the g77 compiler and un-comment out lines for other compilers in these files.

In the case that you use a fortran compiler whose name is not written in these files, write your fortran compiler name into these files. Adjust the optimize level step by step while checking that the calculation result is not changed. Do not use the optimization option for `PEGS4COM.BAT`, since it needs a very small CPU time. In the case automatic save of a local variable is not a default, specify it as a fortran compile option in `MORTRUN.BAT`, `EGS4RUN.BAT` and `EGS4RUNP.BAT` (See section 1.3.2).

When you install the egs4 system on a PC, only two machine-dependence points are this compiler name (and option) and the difference of separator for namelist input described in section 2.5.2.

2.3.3 Check of mortran3 (Mode 2) using check77.mor

A short program (10 lines) named `CHECK77.MOR` is a program used to check the normal operation of the Mortran system. If `CHECK77.MOR` is converted into a Fortran program normally and compiled and executed without any problem, you can judge that the Mortran system is set up correctly. To execute `CHECK77.MOR`, type in

```
C:¥EGS4LECT> MORTRUN CHECK77.MOR
```

Be CARFUL concerning the sixth-letter of `MORTRUN`. It is not `A` but `U`! `MORTRUN.BAT` is a DOS-batch file used to convert a program written in Mortran language into a program written in Fortran, and compile and execute.

When the job is terminated, you will see

```
 **** Results of check77.f ***
  THE FIRST TEN INTEGERS AND THEIR SQUARES:

  I=  1 ( ODD INTEGER)    1
  I=  2 (EVEN INTEGER)    4
```

```
I=  3 ( ODD INTEGER)    9
I=  4 (EVEN INTEGER)   16
I=  5 ( ODD INTEGER)   25
I=  6 (EVEN INTEGER)   36
I=  7 ( ODD INTEGER)   49
I=  8 (EVEN INTEGER)   64
I=  9 ( ODD INTEGER)   81
I= 10 (EVEN INTEGER)  100
```

on the display. If you do not obtain this output, you must check MORTJOB.FOR and MORTJOB.MLS using an editor. Check if all the macro and Mortran statements have been converted into Fortran statements.

2.4 Installation of EGS4

No modification is necessary for egs4 files when you install egs4. You just have to check that egs4 files (ex. EGS4.MOR) run without any trouble.

2.4.1 Running EGS4RUN.BAT

You run a sample user's code, UCSAMPL4.MOR, with sample material data, sampl4.dat, using a DOS-batch file EGS4RUN.BAT to check egs4 files run without any trouble.

```
C:¥EGS4LECT> EGS4RUN UCSAMPL4.MOR SAMPL4.DAT
```

If you use g77, please ignore following warning message.

```
mortjob.for:  In block-data ‘_BLOCK_DATA__’:  mortjob.for:468:  warning:
COMMON/ELECIN/ EKELIM, EKE0(10),EKE1(10),CMFP0(10),CMFP1(10),RANGE_
Înitialization of large (108772-unit) aggregate area ‘elecin’ at ()̂
currently very slow and takes lots of memory during g77 compile -- to be
improved in 0.6
```

Wait until the job is finished. The content of EGS4RUN.BAT is shown in Fig. 2.

When the execution of EGS4RUN.BAT is finished, compare the calculation result in MORTJOB.OUT [4] with the corresponding output described in SLAC-265 (shown as Fig. 4). If they are the same, you can judge that EGS4 is installed.

You run EGS4RUN.BAT similarly when you make your own user's code and material data. Use EGS4RUNP.BAT when you use PRESTA. [5]

If the output is not the same as in Fig. 4, or the output is stopped in the middle, or you get an error message, some problem has occurred. You must grasp what has happened and fix the reason for the error. To do this, investigate other output files (see section 4.2 for the contents) and execute the content of EGS4RUN.BAT one by one according next part.

[4] You can change OUT into OU6 or OT6 by changing open statement for unit 6 in UCSAMPL4.MOR.

[5] For example, type in C:¥EGS4LECT> EGS4RUNP UCNAI.MOR NAI.DAT

2.4.2 Execution by individual commands

Skip this part in the case that UCSAMPL4.MOR runs without any trouble. Use this part only when you have some trouble in executing EGS4RUN.BAT, or you want to understand the content of EGS4RUN.BAT.

This part describes how to run user's code UCSAMPL4.MOR using individual DOS commands. The execution of EGS4 is divided into the following three parts:

1. Catinate user's code and EGS4 files into one file.
2. Convert a Mortran file into a Fortran file.
3. Compile and execute a fortran file.

First check if the following open statements are contained in a macro in STEP 1 of UCSAMPL4.MOR. If some lines are missing, add them.

```
open(unit=6 ,file='mortjob.output');
open(unit=8 ,file='mortjob.dummy');
open(unit=12,file='mortjob.xsec');
```

Then check if a line of $OPEN; is written in STEP.2. (This line is a template to be replaced by OPEN statements written in STEP 1.)

Then, catinate the user's code and files of egs4 into one big mortran file named MORTJOB.
MOR. Also copy material data to a file named MORTJOB.XSE.

```
C:¥EGS4LECT>COPY LISTING.ON+EGS4MAC.MOR+NRCC4MAC.MOR+PAIRMAC.MOR+
KEK4MAC.MOR+UCSAMPL4.MOR+KEK4.MOR+EGS4BLOK.MOR+EGS4.MOR MORTJOB.MOR
C:¥EGS4LECT>COPY SAMPL4.DAT MORTJOB.XSE
```

By typing

```
C:¥EGS4LECT> MORTRAN3
```

MORTRAN3.EXE runs. It reads in MORTJOB.MOR and outputs MORTJOB.FOR. Compile and execute this fortran file:

```
C:¥EGS4LECT> lf90 MORTJOB.FOR -nw -lst       (For Lahey Fortran)
C:¥EGS4LECT> FL32 /Ox /W0 /G4 MORTJOB.FOR   (For MS-Fortran/PS)
C:¥EGS4LECT> MORTJOB.EXE
```

Then you should get the same output as shown in Fig. 4 in MORTJOB.OUT.

2.5 Installation of PEGS4

2.5.1 Generation of a PEGS4 executable file

Generate and keep an executable file of PEGS4 using PEGS4COM.BAT. Key in

```
C:¥EGS4LECT> PEGS4COM
```

to compile PEGS4.MOR, and keep the executable file as PEGS4.EXE. The content of PEGS4COM.BAT is shown in Fig. 5. Usually, PEGS4 is not modified afterwards.

If you experience any trouble during the compile, open PEGS4.MOR using an editor to check that CALL IOINIT is commented out. At that time also check that following OPEN statement is contained:

```
OPEN(5,  file='pgs4job.inp');
OPEN(6,  file='pgs4job.lst');
OPEN(7,  file='pgs4job.dat');
OPEN(8,  file='pgs4pepr.dat');
OPEN(9,  file='pgs4form.dat');
OPEN(10, file='pgs4job.err');
OPEN(21, file='pgs4job.plt');
OPEN(22, file='aprime.dat');
```

2.5.2 Test execution of PEGS4

Using PEGS4RUN.BAT, execute PEGS4.EXE to produce material data as a test run. Here, SAMPL4.INP is used as sample input data. You must make input data when you run PEGS4 afterwards.

```
C:¥EGS4LECT> PEGS4RUN SAMPL4.INP PEGS4.OUT
```

Here, the first and second operands specify the input file name and output file names respectively. The content of SAMPL4.INP is shown in Fig. 9. If you use an MS-Fortran/Power Station, change the indicator of the end of the namelist-input from "&END" to "/", as shown in the later half of Fig.9. [6]

2.5.3 Use of EXAMIN.MOR to check the PEGS4 output

PEGS4 automatically sets energy bins and linearly interpolates physical quantities (ex. total cross section, branching ratio et al), then output the coefficients. It is necessary to compare the PEGS4 outputs on a new computer (to install PEGS4) and a standard computer. However, it is sometimes difficult to compare the PEGS4 outputs directly. Due to some reasons, like a difference in the precision, different energy bins may be used on a new computer and a standard one. In that case, identical physical quantities are described quite differently. For this comparison, a EGS4 user's code, EXAMIN.MOR, is used

[6] You also must change the indicator in the same way for any other namelist-input if you use MS-Fortran/ Power Station.

to check the PEGS4 output. EXAMIN.MOR reads in PEGS4 data and outputs them in tabular form so that man can easily read, and then terminate, without calling a shower. To run EXAMIN.MOR, type in,

```
C:¥EGS4LECT> EGS4RUN EXAMIN.MOR PEGS4.OUT
```

Here PRESTA is not used.

Compare newly generated MORTJOB.OUT and EXAMIN.OUT,

```
C:¥EGS4LECT>FC EXAMIN.OUT MORTJOB.OUT
```

If they are the same or differ in the 5th digits several times, PEGS4 on the new computer runs in the same way as PEGS4 on the standard computer. If they differ more apparently, you must investigate the reason. EXAMIN.OUT is an output file from EXAMIN.MOR. Here, the input of EXAMIN.MOR is calculated using PEGS4 with SAMPL4.INP (=input file) and PGS4PEPR.DAT[7] (= Stom and Israel's photoeffect cross section data) on KEK's central computer(ccce5=Hitach's WS). The content of EXAMIN.OUT is shown in Figs. 10 and 11.

[7] On pgs4pepr.dat, see footnote 2 in "Installation to Unix computer" part.

3 Installation to a Macintosh

LS FORTRAN for a Macintosh and a MPW (Macintosh Programmer's Workshop) Shell are supposed to be used. The MPW shell is contained in the LS Fortran package.

3.1 Get the system

From the FTP server of KEK, get the EGS4 system for a Macintosh by anonymous ftp. On a Macintosh, enable "`FTP enable`" and "`MacBinary enable`". [8] Then connect to Node name: `ftp.kek.jp (130.87.34.29)` by ftp and enter "`anonymous`" as the user's ID and "Your ID@node name" as the password to login. [9] Then, change the mode to binary by

```
ftp> binary
```

and get the following files by the `get` command:

`/kek/kek_egs4/egs4macintosh_kek/egs4macintosh.sea`	`(EGS4 system)`
`/kek/kek_egs4/egs4macintosh_kek/user_code.sea`	`(EGS4 sample user's code)`

3.2 Extraction of a file

The obtained files are self-extraction files made by `Drop Stuff`. Copy `egs4macintosh.sea` to the hard-disk of a Macintosh, then double-click the icon to extract files. During extraction, an egs4macintosh folder is first created, then files are extracted into that folder. (You can change the folder name before the folder is created.) Extract files from `user_code.sea` in the same way.

3.3 Installation of MORTRAN

We use LS FORTRAN for a Macintosh and MPW(Macintosh Programmer's Workshop) shell. LS FORTRAN runs on the Shell of MPW. First click the MPW Shell icon in the MPW folder. You can see Worksheet of MPW. We operate in the Worksheet afterwards.

The `MORTRAN` system consist of `mortran3.f` and `mornew77.raw. mornew77.raw` is a data file of rules to convert a Mortran program to a Fortran program. By compiling

[8] These are necessary to get files with the property of self-extraction. If this property is not recognized, individual files can not be extracted. A test was made using NCSA telnet Ver. 2.5.

- It is also possible to use Fetch to get files and file-extraction. Specify "Self-extraction file" in Fetch.
- In the case that you get files via PC or unix, this "self-extraction" property is lost. In this case, individual files can be extracted using StuffIt Expander Ver. 3.5.2 (Ver. 4.0.2 did not work for extraction).

[9] In the case that telnet is used before ftp start, the `ls` command does not work. It is thus necessary to start ftp when telnet is not used.

mortran3.f, you get the executable file mortran3. mortran3 runs in the following two modes:

Mode I: Generation of mortran3.dat mode

> This mode is utilized during Mortran installation. mornew77.raw is converted into Hexagonal expression, and the converted data file is saved as mortran3.dat.

Mode II: Mortran/Fortran conversion mode

> This mode is utilized in the usual EGS4 calculation. Using mortran3.dat, mortran3 convert a file written in Mortran language into a file written in Fortran language.

3.3.1 Generation of mortran3.dat(Mode I, machine dependence)

Compile mortran3.f without any change, since mortran3.f is set as MODE-I when it is distributed. On a worksheet, type in

```
RunPPC  mortran3.f   -saveall  (enter)
```

to compile mortran3.f, and execute mortran3 and to get mortran3.dat. Here, the compiler option "-saveall" must be specified in order to specify to save local variables. Be careful to push enter (NOT Return) after typing command. Also, be careful not to shorten the file name. (The extension of ".f" at the last of the file name is necessary.)

The first line of the obtained mortran3.dat,

```
v....2C USER F77 11JUN85
```

should be erased. You should erase all of this line. (BE CARFULL NOT TO LEAVE A BLANK LINE AT THE TOP OF THE FILE!)

If you do not obtain mortran3.dat, open mortran3.f by an editor to check if

```
IMODE=1
```

is written. If you find line of CALL IOINIT, comment out that line, since it does not work on machines other than a SUN.

Once mortran3.dat is created during the installation, mortran3 is never used in Mode-I afterwards. mortran3 is changed into Mode-II for the usual egs4 calculation. Open mortran3.f by the editor, and change the line IMODE=1 to

```
IMODE=2
```

Then compile again.

```
RunPPC mortran3.f -saveall (enter)
```

The newly obtained mortran3 runs in Mode-II.

3.3.2 Modification of the compiler option (machine dependence)

Please skip this part when you use LS Fortran.

Read the corresponding part of section 1.3.2 or section 2.3.2 when you use a compiler different from LS-Fortran. The actual machine dependence is not yet known, since only LS-Fortran has been tested on a Macintosh so far.

3.3.3 Check of mortran3 (Mode 2) using check77.mor

`check77.mor` is a short program (10 lines) that is used to check the normal operation of the Mortran system. If `check77.mor` is converted into a Fortran program normally, and compiled and executed without any problem, you can judge that the Mortran system is set up correctly.

To execute `check77.mor`, type in

```
mortrun check77.mor
```

Be CARFUL concerning the sixth-letter of `mortrun`. It is not a but u! Do not forget to push "`enter`" as the last step.

`mortrun` is a MPW-shell script that is used to convert a program written in Mortran language into a program written in Fortran, and to compile and execute. The contents of `mortrun` are:

```
echo "mortrun has started"
delete mortjob.≈|| echo "There aren't any mortjob.≈"
catenate {1} >mortjob.mortran
mortran3
alert 'Wait until mortran3 finishes'
RunPPC mortjob.f -saveall
```

The "≈" mark means wild-card. "`alert 'Wait until mortran3 finishes'`" prevents execution of "`RunPPC mortjob.f -saveall`" before `motran3` finishes.
If "`delete mortjob.≈`" is executed and there are no file matches to "`mortjob.≈`", the computer runs abnormally and stops (It hangs.) To prevent this hang up, a dummy command is written after the `delete` command. The command after "`||`" is executed in the case that the `delete` command fails.

When the job starts, you hear a beep-sound, and the following line is indicated:

```
Wait until mortran3 finishes
```

While `mortran3` is running, a diamond-shaped icon is indicated at the right-top corner of the display. When the execution of `mortran3` finished,

```
Program  executed  STOP  statement
```

is indicated. Click "OK" in this statement window. Then click "OK" for

```
Wait until mortran3 finishes
```

Then `RunPPC mortjob.f -saveall` is executed and you will see

```
  **** Results of check77.f ***
   THE FIRST TEN INTEGERS AND THEIR SQUARES:

   I=  1 ( ODD INTEGER)    1
   I=  2 (EVEN INTEGER)    4
   I=  3 ( ODD INTEGER)    9
   I=  4 (EVEN INTEGER)   16
   I=  5 ( ODD INTEGER)   25
   I=  6 (EVEN INTEGER)   36
   I=  7 ( ODD INTEGER)   49
   I=  8 (EVEN INTEGER)   64
   I=  9 ( ODD INTEGER)   81
   I= 10 (EVEN INTEGER)  100
```

in a new window on the display. A diamond-shaped icon is displayed on the right-top corner of the display to indicate that `mortjob.exe` is active.

If you do not obtain this output, you must check `mortjob.f` and `mortjob.list` using the editor. Check if all the macro and mortran statement are converted into Fortran statements.

If you type in only `mortrun` and push `enter` by mistake, `mortrun` stops and "`catenate`" is indicated in the left-top corner. This happens because the `catenate` command can not find a file to be copied. To escape from this situation, push : Apple key (A symbol like # with rounded corners. Or a symbol like combination of 4 tennis racket. Usually, you find it at both sides of the space key.) and "." (Period key) simultaneously. Then,

```
### MPW Shell - Execution of mortrun terminated.
```

will be shown on the worksheet and the script is terminated.

3.4 Installation of EGS4

No modification is necessary for egs4 files when you install egs4. You just have to check that egs4 files (ex. `egs4.mor`) runs without any trouble.

3.4.1 Running egs4run

You run `ucsampl4.mor` (user's code) using `egs4run` to check that egs4 runs without any trouble. `egs4run` is a MPW-shell script which is used to run egs4 (See Fig. 3). As material data, `sampl4.dat` is used. Execute `egs4run`,

```
egs4run ucsampl4.mor sampl4.dat (enter)
```

If you do not specify the file name, the script stops at the line of catenate in the same way as in `mortrun`. In that case push :Apple key (A symbol like # with rounded corners.

Or a symbol like combination of 4 tennis racket. Usually, you find it at both sides of the space key.) and "." (Period key) simultaneously.

When the job starts, you hear a beep-sound and the following line is indicated:

```
Wait until mortran3 finishes
```

While `mortran3` is running, a diamond-shaped icon is indicated at the right-top corner of the display. When the execution of `mortran3` finished,

```
Program  executed  STOP  statement
```

is indicated. Click "OK" in this statement window. Then click "OK" for

```
Wait until mortran3 finishes
```

Then, `RunPPC mortjob.f -saveall` is executed. Wait until `mortjob` is finished.

The calculation result is output in `mortjob.output`. Compare this with the output of `ucsampl4.mor` described in Fig.4 (This output is identical to that described in SLAC-265.) If both are the same, you can judge that the installation of EGS4 has been performed correctly.

You run `egs4run` similarly when you make your own user's code and material data. In the case that you use PRESTA, use `egs4runp` instead of `egs4run`. [10]

If the output is not the same as in Fig. 4 or the output has stopped in the middle, or you get an error message, some problem has occurred. You must grasp what has happened and fix the reason for the error. To do this, investigate other output files (see section 4.2 for the contents) and execute the content of `egs4run` one by one according to the next part.

3.4.2 Execution by individual commands

Skip this part in the case that `ucsampl4.mor` runs without any trouble. Do this part only when you have some trouble in executing `egs4run`, or you want to understand the contents of `egs4run`.

This part describes how to run user's code `ucsampl4.mor` using individual MPW-Shell commands. The execution of `EGS4` is divided into three parts:

1. Catinate user's code and `EGS4` files into one file.

2. Convert a Mortran file into a Fortran file.

3. Compile and execute a fortran file.

First check whether the following `open` statements are contained in a macro in STEP 1 of `uc`
`sampl4.mor`. If some lines are missing, add them.

[10] For example, type in `egs4runp ucnai.mor nai.dat`

```
open(unit=6 ,file='mortjob.output',status='new');
open(unit=8 ,file='mortjob.dummy',status='new');
open(unit=12,file='mortjob.xsec',status='old');
```

Then check if a line of `$OPEN;` is written in STEP.2. (This line is a template to be replaced by OPEN statements written in STEP 1.)

Then catinate the user's code and files of `egs4` into one big mortran file named `mortjob.`
`mortran`. Also copy material data to a file named `mortjob.xsec`. (Do not forget to push `enter` after each command!)

```
catenate listing.on > mortjob.mortran
catenate egs4mac.mortran >>mortjob.mortran
catenate nrcc4mac.mortran >>mortjob.mortran
catenate kek4mac.mortran >>mortjob.mortran
catenate ucsampl4.mor >>mortjob.mortran
catenate kek4.mortran >> mortjob.mortran
catenate egs4blok.mortran >> mortjob.mortran
catenate egs4.mortran >>mortjob.mortran

catenate sampl4.dat >mortjob.xsec
```

By typing

```
mortran3
```

`mortran3` runs. It reads in `mortjob.mortran` and outputs `mortjob.f`. Compile and execute this fortran file,

```
RunPPC mortjob.f -saveall -opt=3
```

You should then get the same output in `mortjob.output` as shown in Fig. 4.

3.5 Installation of PEGS4

3.5.1 Generation of PEGS4 executable file

Generate and keep executable file of PEGS4 using `pegs4compile` . Key in

```
pegs4compile
```

to compile `pegs4.mortran` and keep the executable file as `pegs4.exe`. The content of `pegs4compile` is shown in Fig. 7. Usually, PEGS4 is not modified afterwards.

If you experience any trouble during compiling, open `pegs4.mortran` using an editor to check that `CALL IOINIT` is commented out. At that time also check that the following OPEN statement is contained:

```
OPEN(5,  file='pgs4job.pegs4inp');
OPEN(6,  file='pgs4job.pegs4lst');
OPEN(7,  file='pgs4job.pegs4dat');
OPEN(8,  file='pgs4pepr.dat');
OPEN(9,  file='pgs4form.dat');
OPEN(10, file='pgs4job.pegs4err');
OPEN(21, file='pgs4job.pegs4plot');
OPEN(22, file='aprime.data');
```

3.5.2 Test execution of PEGS4

Using `pegs4run`, execute `pegs4.exe` to produce material data as a test run. Here, `sampl4.inp` is used as sample input data. You must make input data when running PEGS4 afterwards.

```
pegs4run sampl4.inp pegs4.output
```

Here, the first and second operands specify the input file name, and output file name, respectively. The contents of `pegs4run` and `sampl4.inp` are shown in Figs. 8 and 9, respectively. Here, if you forget to specify the file name, `pegs4run` stops. In this case, push Apple key and "." key simultaneously to terminate `pegs4run`.

3.5.3 Use of examin.mor to check the PEGS4 output

PEGS4 automatically sets energy bins, linearly interpolates physical quantities (ex. total cross section, branching ratio et al) and then outputs the coefficients. It is necessary to compare the PEGS4 outputs on a new computer (to install PEGS4) and a standard computer. However, it is sometimes difficult to compare PEGS4 outputs directly. Due to some reasons, like a difference of precision, different energy bins may be used on a new computer and a standard one. In that case, identical physical quantities are described quite differently. For this comparison, the EGS4 user's code `examin.mor` is used to check the PEGS4 output. `examin.mor` reads in PEGS4 data and outputs them in tabular form so that man can easily read and then terminate without calling a shower. To run `examin.mor`, type in

```
egs4run examin.mor pegs4.output
```

Here, `PRESTA` is not used.

Compare the newly generated `mortjob.output6` and `examin.out`,

```
compare mortjob.output6 examin.out
```

If they are the same, or differ in the 5th digits several times, PEGS4 on the new computer runs in the same way as PEGS4 on a standard computer. If they differ more apparently, you must investigate the reason. `examin.out` is an output file from `examin.mortran`. Here, the input of `examin.mor` is calculated using PEGS4 with `sampl4.inp` (=input file) and `pgs4pepr.dat`[11] (= Stom and Israel's photoeffect cross section data) on KEK's central computer (ccce5=Hitach's WS). The content of `examin.out` is shown in Figs. 10 and 11.

[11] On `pgs4pepr.dat`, see footnote 2 in "Installation to Unix computer" part.

4 Exercise problem

4.1 Notice points in the exercise

- If you use Emacs, set the display by typing in

```
% setenv DISPLAY <ip address>:0.0
```

- Please ignore any IEEE error generated while PEGS4 is running.
- If you perform the exercise using a PC at KEK, please not to operate files outside of the specified directory. Please do not to change the environment of Windows.

4.2 List of files created by EGS4RUN

After EGS4RUN runs, the following files are created:

File name on PC	Ext. on unix and Mac	Content
MORTJOB.MOR	.mortran	User's code+ EGS4 (MORTRAN Language)
MORTJOB.XSE	.xsec	Duplicate of material data file
MORTJOB.MLT	.list	Listing of Mortran file Error info. during Mortran → Fortran conversion
MORTJOB.FOR	.f	Fortran Source code
MORTJOB.OUT	.output(6)	Calculation result
MORTJOB.DUM	.dummy	Echo-back of material data by HATCH
MORTJOB.LST		Listing of Fortran file Error info. during Fortran compilation
MORTJOB.OBJ		Object File
MORTJOB.MAP		Map file from Linker
MORTJOB.EXE	(a.out)	Executable file

If you fail to compile, check the following files:

- MORTJOB.MOR: Is user's code included?
- MORTJOB.MLT: Are any Mortran errors generated?
- MORTJOB.FOR: Isn't any macro statement left in the FORTRAN source?
- MORTJOB.LST: Are any fortran errors generated?

In the case that you experience trouble while reading in material data, you can find which part of the material data is read in correctly by investigating `MORTJOB.DUM`.

On a unix computer, Fortran errors are indicated on the display only, and are not written in a file. Thus, in the case that you notice a fortran error during fortran compiling, stop the job immediately by pushing the "cntl" key and "z" key simultaneously, and write down the line number. (Or, you can record error message in a file by adding ">& file name" to a statement of f77 or g77 in mortrun or egs4run.)

4.3 Exercise problem of MORTRAN

1. Make a program to output prime numbers under 50. (Start programming from `check77.mor`.)

   (Hint)

   - Use MORTRAN usage for specifying a DO-loop area by "[" and "]".
   - `N=MOD(I,J);` will output the rest when you divide I by J. If N$\neq$0 for any J of J<I, I is a prime number.
   - Do not use the tab, since it causes a MORTRAN error.

2. Calculation of π

   (a) Uniformly set points within a regular square of $-1 < x < 1$, $-1 < y < 1$ by using sets of two random numbers.

   (Hint)

   - Copy the following macro to generate a random number from `ucsampl4.mor`. This macro generates the random number between 0 and 1. Use this macro.

     ```
     REPLACE {$RANDOMSET#;} WITH
     {IXX=IXX*663608941;{P1}=IXX*0.23283064E-09;
     IF(IXX.LT.0){P1}={P1}+1.0;}
     ```

     You do not need to write this macro in the EGS4 user's code, since the equivalent macro is automatically read in from `nrcc4mac.mor` when you run EGS4.

   - Copy the following line for setting the initial value of a random number.

     ```
     IXX=987654321;  "RANDOM NUMBER GENERATOR SEED"
     ```

   - By writing `$RANDOMSET` <Variable name>;, you obtain a random number between 0 and 1 as the value of <Variable name>.

   (b) Calculate the ratio of the number of points inside a circle of $x^2 + y^2 = 1$ to the total number of points above. Calculate the value of π from this ratio.

3. Point isotropic source

   (a) Set points uniformly inside a cubic of $-1 < x < 1$,$-1 < y < 1$ and $-1 < z < 1$ by generating sets of 3 random numbers.

(b) Points inside a sphere of $x^2 + y^2 + z^2 = 1$ distributes uniformly from the origin point. By dividing the coordinates of these points inside this sphere by the distance from the origin to the point, you obtain the direction cosine of a point isotropic source. Obtain ten sets of direction cosines of this point isotropic source.

4.4 Exercise problem of EGS4 and PEGS4

1. Calculation of a NaI detector

   Change `ucnai3p.mor` as follows, and see how the peak efficiency and total efficiency are modified:

   (a) Double the thickness of the effective volume.

   (b) Change the source energy to 667 keV (^{137}Cs). (The thickness of the effective volume is returned to the original value.)

   (c) Change the source from one photon energy source (1.33 MeV) to two photon energy sources (1.173 MeV and 1.333 MeV, the intensity ratio is 1:1).

   (Hint) Generate a random number (R) inside the `CALL SHOWER` loop. Then select the photon energy depending on R. For example, select 1.173 MeV when $0 < R < 0.5$, or select 1.333 MeV when $0.5 < R < 1$.

   (d) Change the source direction from the Z-direction to isotropic.

   (Hint) Copy the isotropic source made in the Mortran exercise to inside of `CALL SHOWER`-loop.

2. Calculation of a Ge detector

   (a) Make material data of Ge using PEGS4. The energy range is 10 keV–50 MeV for γ and 521 keV–50 MeV for e^-.

   (Hint) Copy `sampl4.inp` and change any necessary points.

   (b) Change material of a sensitive area in `ucnai3p.mor` from NaI to Ge.

   (c) Calculate the response function of the Ge detector using results of (a) and (b). Compare the NaI detector and Ge detector with the same effective-area sizes. Which one has a larger peak efficiency and total efficiency?

   (hint) Material data of Al are necessary to calculate the Ge detector. Thus, add either Al material data or whole `nai.dat` after material data of Ge using command or the editor. (ex. `cat ge.dat nai.dat > ge-al.dat` on unix.)

3. Calculation of an air ion chamber

   (a) Make material data of air (20°、 1 atm) while specifying `IAPRIM=1` in PEGS4 input. Use the same energy ranges as that in 2.(a).

(b) Change the sensitive area in `ucnai3p.mor` to air.

(c) Calculate the response of the air ionization chamber using the material data of air, and estimate the fraction of energy deposited in air to the total incident energy.

(Hint) The output of energy deposition summary.

(d) Using the W value of air (33.8 eV/pair), calculate the output of this air ionization chamber (Coulomb/source) for one incident photon of 1.33 MeV.

(Hint) If you divide the energy deposited in air by the W-value and multiply by the electron charge magnitude (1.602×10^{-19} C/e), you obtain the output of the air ionization chamber in Coulomb/incident.

4.5 Answer for MORTRAN exercise

1. Mortran program to output a prime number under 50.

```
!COMMENTS;!INDENT C5;!INDENT M4;!INDENT F2;!LIST;
DO I=1,50 [
  IPRIME=1;
  DO J=2,I-1 [
    K=MOD(I,J);
    IF(K.EQ.0)IPRIME=0; "Not a prime number"
    ]
  IF(IPRIME.EQ.1) OUTPUT I; (I3,' is a prime number.');
  ]
STOP;
END;
%%
```

2. Calculation of π. Program for (a) and (b) is shown.

```
!COMMENTS;!INDENT C5;!INDENT M4;!INDENT F2;!LIST;
REPLACE {;COMIN/RANDOM/;} WITH {;COMMON/RANDOM/IXX;}
REPLACE {$RANDOMSET#;} WITH
{IXX=IXX*663608941;{P1}=IXX*0.23283064E-09;IF(IXX.LT.0){P1}={P1}+1.;}
;COMIN/RANDOM/;
IXX=123456789;
INSIDE=0;
NCASES=100000;
DO I=1,NCASES [
  $RANDOMSET X1; X2=X1*2.0-1.0;
  $RANDOMSET Y1; Y2=Y1*2.0-1.0;
  IF(X2*X2+Y2*Y2.LE.1.0)[
    INSIDE=INSIDE+1;
    ]
  ]
PIVAL=4.0*FLOAT(INSIDE)/FLOAT(NCASES);
OUTPUT PIVAL; (' Estimated pi value='1PE12.5);
STOP;
END;
%%
```

3. Point isotropic source. Program for (a) and (b) is shown.

```
!COMMENTS;!INDENT C5;!INDENT M4;!INDENT F2;!LIST;
REPLACE {;COMIN/RANDOM/;} WITH {;COMMON/RANDOM/IXX;}
REPLACE {$RANDOMSET#;} WITH
{IXX=IXX*663608941;{P1}=IXX*0.23283064E-09;IF(IXX.LT.0){P1}={P1}+1.;}

;COMIN/RANDOM/;
IXX=123456789;
NCASES=10;
DO I=1,NCASES [
  :Sampling_start: "Label for goto statement"
  $RANDOMSET X1; X2=X1*2.0-1.0;
  $RANDOMSET Y1; Y2=Y1*2.0-1.0;
  $RANDOMSET Z1; Z2=Z1*2.0-1.0;
  R2=SQRT(X2*X2+Y2*Y2+Z2*Z2);
  IF(R2.GT.1.0) GOTO :Sampling_start:;
  UI=X2/R2; VI=Y2/R2; WI=Z2/R2;
  OUTPUT UI,VI,WI; (' UI,VI,WI=',3E11.4);
  ]
STOP;
END;
%%
```

4.6 Answer for EGS4 and PEGS4 exercise

1. Modification of `ucnai3p.mor`

   (a) The peak and total efficiencies are 35.824% and 72.392%, respectively. After the sensitive thickness is doubled, the peak and total efficiencies are 52.039% and 92.137%, respectively. Change one line to

   ```
   TDE=15.24;
   ```

   (b) The peak and total efficiencies are 58.294% and 85.176%, respectively. Change one line to

   ```
   EI=0.667;
   ```

   (c) The peak and total efficiencies are 36.647% and 74.255%, respectively. Change the beginning part of `CALL SHOWER`-loop to

   ```
   DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"
   $RANDOMSET AA;
   IF(AA.LT.0.5) [EI=1.33 +ABS(IQI)*PRM;] "TOTAL ENERGY OF PARTICLE"
   ELSE [EI=1.17 +ABS(IQI)*PRM;]
   ```

   Here, `ECNSV` does not operate properly for a multiple energy source, like in this example. (The sum of the energy loss is 0.9397. It should be 1 correctly.) To make `ECNSV` operate properly, change the scoring of the total kinetic energy as follows:

   ```
   " STEP 6"
   TOTKE=0.0;

   "STEP 7"
   "After EI is determined"
   AVAILP=EI + IQI*PRM;  "AVAILABLE K.E. (MeV) (MUST BE REAL*8)"

   "After shower call"
   TOTKE=TOTKE+AVAILP; "TOTAL (AVAILABLE) K.E."
   ```

   (d) The peak and total efficiencies are 7.2549% and 21.510%, respectively. The finite solid angle of the sensitive area seen from the source affects the results. Change the beginning part of `CALL SHOWER`-loop as follows:

```
DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"
"Point isotropic source"
:Sampling_start: "Label for goto statement"
$RANDOMSET X1; X2=X1*2.0-1.0;
$RANDOMSET Y1; Y2=Y1*2.0-1.0;
$RANDOMSET Z1; Z2=Z1*2.0-1.0;
R2=SQRT(X2*X2+Y2*Y2+Z2*Z2);
IF(R2.GT.1.0) GOTO :Sampling_start:;
UI=X2/R2; VI=Y2/R2; WI=Z2/R2;
```

2. Calculation of Ge detector

   (a) Execute `PEGS4` using following input data,

   ```
   ELEM
    &INP IRAYL=1 &END
   GE                            GE
   GE
   ENER
    &INP AE=0.521,UE=50.0,AP=.01,UP=50.0 &END
   TEST
    &INP &END
   PWLF
    &INP &END
   DECK
    &INP &END
   ```

   (b) Change a line of the user's code as follows:

   From,

   ```
   DATA MEDARR/$S'NAI-IAPRIM',14*' ',
   ```

   to

   ```
   DATA MEDARR/$S'GE',22*' ',
   ```

   From,

   ```
   IEDGFL(8)=53;  "53:Atomic number of I"
   ```

   to,

   ```
   IEDGFL(8)=32;  "32:Atomic number of Ge"
   ```

   Adjust the length of blanks after the material name (ex. `22*' '` or `14*' '`) so that sum of the length of material name and the length of blanks are equal to 24-letters. If you do not adjust this properly, the head of second material name does not come to the head of the dimension. This will destroy the proper execution of EGS4-job.

   (c) The peak and total efficiencies are 37.529% and 85.137%, respectively. (The total efficiency increases due to a larger density of sensitive area. However, the atomic number of Ge is smaller than effective atomic number of NaI. This counters the effect of a larger density. Thus, the change in the peak efficiency is smaller than that of the total efficiency.)

3. Calculation of the air ionization chamber

   (a) Execute PEGS4 with the following input:

```
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
ENER
 &INP AE=0.521,UE=50.0,AP=.01,UP=50.0 &END
TEST
 &INP &END
PWLF
 &INP &END
DECK
 &INP &END
```

   (b) Change 2 lines of the user's code:

```
DATA MEDARR/$S'AIR',21*' ',
```

```
IEDGFL(8)=18;  "18:Atomic number of Ar"
```

   (c) The answer is the energy deposition of all particles in region 8. It is 1.268×10^{-4}.

   (d) $1.33 \times 10^6(\mathrm{eV/source}) \cdot 1.268 \times 10^{-4} \cdot \frac{1}{33.8(\mathrm{eV/pair})} \cdot 1.6 \times 10^{-19}(\mathrm{Coulomb/pair})$
   $= 7.98 \times 10^{-19}(\mathrm{Coulomb/source})$

   - Most of the energy deposition in the sensitive area of air is a contribution of electrons generated in a cap made of Al. In the case of most of real ion chambers, either a cap is not used, or the cap is made by a material whose atomic number is close to that of air.
   - By using a variable `LATCH` in `COMMON/STACK`, it is possible to record the region and interaction when a photon is converted to an electron and to score electrons while distinguishing the region and the interaction of the conversion. See Tutor 5 in SLAC-265 for detail concerning `LATCH`.
   - All of the photoeffects in air were assumed to be that of Ar here. (See `IEDGFL(8) =18;` above.) Ar is the heaviest atom among the considered contents of air. This assumption is usually employed in the EGS4 calculation, because it is impossible to calculate X-rays from more than one element in a compound or mixture unless a special program is added to EGS4. See A. Del. Guerra, Nucl. Instrum. and Meth **A 306**, 378-385 (1991) for this special program.
   - Due to a large statistical error, a result with a 10-20% difference may be obtained on different computers.

(file: egsins7.tex, egsins7a.tex and egsins7b.tex)

Figure 1: List of egs4run (unix)

```
# * E G S 4 R U N *
# Usage:  egs4run $1 $2
#         $1: User code name.  ex. ucsampl4.mor
#         $2: Material data file.  ex. sampl4.dat

echo "egs4run for nrcc version started"
echo "user/egs4 configuration entered"; echo " "
rm mortjob.*
rm a.out
echo "Do you want to include MACRO in the mortran list ?
(yes or no. null means yes.)"
read key
case $key in
   no ) cp egs4mac.mortran      mortjob.mortran
        echo "MACRO is not included in mortran list";;
   *  ) cp listing.on           mortjob.mortran
        cat egs4mac.mortran   >>   mortjob.mortran
        echo "MACRO is included in mortran list";;
esac
echo "Do you use PRESTA ? (yes or no. null means yes.)"
read key
case $key in
   no ) cat nrcc4mac.mortran >>   mortjob.mortran
        echo "Default EGS4 is selected";;
   *  ) cat nrc4macp.mortran >>   mortjob.mortran
        cat presta.mortran   >>   mortjob.mortran
        echo "PRESTA is selected";;
esac
cat pairmac.mortran >> mortjob.mortran
cat sunmacs.mortran >> mortjob.mortran
cat kek4mac.mortran >> mortjob.mortran
if test $1
   then cat $1                >> mortjob.mortran   #USER CODE is joined.
   else echo "Key in user code"
        read ucode
        cat $ucode             >>        mortjob.mortran
fi
cat kek4.mortran   >>     mortjob.mortran
cat egs4blok.mortran >> mortjob.mortran
cat egs4.mortran    >> mortjob.mortran
if test $2
   then cp  $2           mortjob.xsec      #Xsec data is copied
   else echo "Key in data-file"
        read dfile
        cp $dfile         mortjob.xsec     #Xsec data is copied.
fi

echo "mortran3 procedure entered"; echo " "
mortran3
echo "f77 procedure entered"; echo " "
f77 mortjob.f
echo "a.out procedure entered"; echo " "
time a.out &
echo "egs4run procedure finished"
```

Figure 2: List of egs4run (PC)

```
REM Usage: EGS4RUN %1 %2
REM %1: User Code (ex: ucsampl4.mor)
REM %2: Material Data (ex:sampl4.dat)

DEL MORTJOB.*

REM Produce MORTJOB.MOR.
COPY LISTING.ON+EGS4MAC.MOR+NRCC4MAC.MOR+PAIRMAC.MOR+KEK4MAC.MOR+%1+KEK4.MOR+
EGS4BLOK.MOR+EGS4.MOR MORTJOB.MOR

REM Copy material data.
COPY %2 MORTJOB.XSE

REM Convert from MORTJOB.MOR to MORTJOB.FOR.
MORTRAN3

REM Compile, link and execute MORTJOB.FOR.
REM Uncomment out next line for Lahey Fortran 90
lf90 MORTJOB.FOR -nw -lst
REM Microsoft FORTRAN POWERSTATION
REM FL32 /Ox /W0 /G4 MORTJOB.FOR

MORTJOB
```

Figure 3: List of egs4run (Macintosh)

```
echo "egs4run has started"
delete mortjob.≈|| echo "There aren't any mortjob.≈"

catenate listing.on > mortjob.mortran
catenate egs4mac.mortran >>mortjob.mortran
catenate nrcc4mac.mortran >>mortjob.mortran
catenate pairmac.mortran >>mortjob.mortran
catenate kek4mac.mortran >>mortjob.mortran
catenate {1} >>mortjob.mortran
catenate kek4.mortran >> mortjob.mortran
catenate egs4blok.mortran >> mortjob.mortran
catenate egs4.mortran >>mortjob.mortran

catenate {2} >mortjob.xsec

mortran3
alert 'Wait until mortran3 finishes'
RunPPC mortjob.f -saveall -opt=3
```

Figure 4: Output of ucsampl4.mor (Unix, PC and Macintosh)

```
 BEFORE HATCH CALL IT0=       0
 EGS SUCCESSFULLY 'HATCHED' FOR     2 MEDIA.
 AFTER HATCH CALL IT1=       0
 Elapsed Time (sec)= 0.00000E+00
1
 SHOWER RESULTS:


       E              Z              W          IQ    IR  IARG

   1000.000      0.0000000       1.000000       -1     2    -1
   163.9947       3.000000      0.9999890        0     3     3
   504.8928       3.000000      0.9987407        0     3     3
   1000.000      0.0000000       1.000000       -1     2    -1
   155.8518       3.000000      0.9998820        0     3     3
   1000.000      0.0000000       1.000000       -1     2    -1
   140.0282       3.000000      0.9987389        0     3     3
   126.1480       3.000000      0.9998272        0     3     3
   202.7643       3.000000      0.9996023        0     3     3
   1000.000      0.0000000       1.000000       -1     2    -1
   116.1481       3.000000      0.9994835        0     3     3
   139.3819       3.000000      0.9988134        0     3     3
   1000.000      0.0000000       1.000000       -1     2    -1
   107.5314       3.000000      0.9994128        0     3     3
   1000.000      0.0000000       1.000000       -1     2    -1

 INCIDENT TOTAL ENERGY OF ELECTRON=      1000.0 MEV
 IRON SLAB THICKNESS= 3.000 CM
 NUMBER OF CASES IN RUN=     10
 LAST RANDOM NUMBER=  -687420815

 ENERGY DEPOSITION SUMMARY:


 FRACTION IN REGION  1= 0.0000000
 FRACTION IN REGION  2= 0.3735966
 FRACTION IN REGION  3= 0.6264034


 TOTAL ENERGY FRACTION IN RUN=   1.000000
   WHICH SHOULD BE CLOSE TO UNITY
 AFTER CALCULATION FINISHED IT2=       0
 Elapsed Time (sec)= 0.00000E+00
```

This output is a result of calculation using egs4.mor of Ver-2002-7-30-1600 KEK or after. If you use older version of egs4.mor, LAST RANDOM NUMBER=1903435093 and several other points are different.

Figure 5: List of pegs4compile (Unix)

```
rm mortjob.*
cp pegs4.mortran mortjob.mortran
echo "mortran3 procedure entered"; echo""
mortran3
echo "f77 procedure entered"; echo""
f77 mortjob.f
mv a.out pegs4.exe
echo "pegs4compile procedure finished"
```

Figure 6: List of PEGS4COM.BAT (PC)

```
del mortjob.*
copy  pegs4.mor mortjob.mor
echo "mortran3 procedure entered"; echo " "
mortran3

echo "Fortran compile procedure entered"; echo " "
REM following two lines are for Lahey F77L-EM/32
f77l3 MORTJOB.FOR /H
386link mortjob -symbol
REM following is for Lahey fortran90
REM lf90  MORTJOB.FOR -nw -lst
REM following is for Microsoft Fortran Power station
REM FL32 /W0 MORTJOB.FOR

ren mortjob.exe pegs4.exe
echo "pegs4.compile procedure finished"
```

Figure 7: List of pegs4compile (Macintosh)

```
echo "pegs4compile has started"
delete mortjob.≈|| echo "There aren't any mortjob.≈"
catenate pegs4.mortran > mortjob.mortran

echo "mortran3 procedure entered"; echo " "
mortran3
alert 'Wait until mortran3 finishes'
echo "fortran.ppc procedure entered"; echo " "
fortran.ppc mortjob.f -saveall -opt=3

linkfortranppc pegs4.exe mortjob.f.o
echo "pegs4.compile procedure has finished"
```

Figure 8: List of pegs4run (Macintosh)

```
echo "pegs4run procedure has started"; echo " "
delete pgs4job.≈|| echo "There aren't any pgs4job.≈"

catenate {1} > pgs4job.pegs4inp

echo "pegs4.exe has started"
pegs4.exe
alert 'Wait until pegs4.exe finishes'
echo "pegs4.exe has finished."
catenate pgs4job.pegs4dat > {2}

echo "pgs4run procedure has finished"
```

Figure 9: List of sampl4.inp

Unix, PC(Lahey) and Macintosh

```
ELEM
&INP IRAYL=1 &END
FE                            FE
FE
ENER
&INP AE=1.5,UE=20000.,AP=0.1,UP=20000. &END
TEST
&INP &END
PWLF
&INP &END
DECK
&INP &END
```

PC(MS-Fortran/Power station)

```
ELEM
&INP IRAYL=1 /
FE                            FE
FE
ENER
&INP AE=1.5,UE=20000.,AP=0.1,UP=20000. /
TEST
&INP /
PWLF
&INP /
DECK
&INP /
```

```
OCALL HATCH
 RAYLEIGH DATA AVAILABLE FOR MEDIUM  1 BUT OPTION NOT REQUESTED.

 EGS SUCCESSFULLY 'HATCHED' FOR ONE MEDIUM.
 CALL TO HATCH COMPLETED

1         SUMMARY OF PHOTON DATA STORED FOR MEDIUM=FE-RAYLEIG
OELECTRON KINETIC ENERGY RANGE:  0.989   19999.488 MEV  PHOTON ENERGY RANGE: 0.100 20000.000 MEV
ORADIATION LENGTH=      1.76084 CM                                  DENSITY=       7.8600 G/CM**3

PHOTON ENERGY   GAMMA MEAN      PHOTOELECTRIC     COMPTON        PAIR                   TOTAL

                FREE PATH(cm)                                                   cm**2/G         cm**-1


    0.100         0.3744           0.59346        0.40654        0.0            3.398E-01      2.671E+00
    0.125         0.5482           0.43702        0.56298        0.0            2.321E-01      1.824E+00
    0.150         0.6984           0.31741        0.68259        0.0            1.822E-01      1.432E+00
    0.200         0.9215           0.17492        0.82508        0.0            1.381E-01      1.085E+00
    0.300         1.1971           0.06787        0.93213        0.0            1.063E-01      8.353E-01
    0.400         1.3837           0.03478        0.96522        0.0            9.195E-02      7.227E-01
    0.500         1.5365           0.02105        0.97895        0.0            8.280E-02      6.508E-01
    0.600         1.6724           0.01456        0.98544        0.0            7.608E-02      5.980E-01
    0.700         1.7975           0.01086        0.98914        0.0            7.078E-02      5.563E-01
    0.800         1.9158           0.00852        0.99148        0.0            6.641E-02      5.220E-01

    1.000         2.1367           0.00587        0.99413        0.0            5.954E-02      4.680E-01
    1.250         2.3847           0.00426        0.99222        0.0035         5.335E-02      4.193E-01
    1.500         2.6152           0.00332        0.98847        0.0082         4.865E-02      3.824E-01
    2.000         2.9910           0.00234        0.96428        0.0334         4.254E-02      3.343E-01
    3.000         3.5194           0.00152        0.89238        0.1061         3.615E-02      2.841E-01
    4.000         3.8473           0.00113        0.81360        0.1853         3.307E-02      2.599E-01
    5.000         4.0505           0.00090        0.73966        0.2594         3.141E-02      2.469E-01
    6.000         4.1703           0.00074        0.67312        0.3261         3.051E-02      2.398E-01
    7.000         4.2389           0.00062        0.61473        0.3846         3.001E-02      2.359E-01
    8.000         4.2737           0.00053        0.56420        0.4353         2.977E-02      2.340E-01

   10.000         4.2776           0.00017        0.48090        0.5189         2.974E-02      2.338E-01
   12.500         4.2181           0.00000        0.40219        0.5978         3.016E-02      2.371E-01
   15.000         4.1374           0.00000        0.34410        0.6559         3.075E-02      2.417E-01
   20.000         3.9642           0.0            0.26451        0.7355         3.209E-02      2.523E-01
   30.000         3.6847           0.0            0.17882        0.8212         3.453E-02      2.714E-01
   40.000         3.4782           0.0            0.13400        0.8660         3.658E-02      2.875E-01
   50.000         3.3234           0.0            0.10681        0.8932         3.828E-02      3.009E-01
   60.000         3.2029           0.0            0.08862        0.9114         3.972E-02      3.122E-01
   70.000         3.1130           0.0            0.07577        0.9242         4.087E-02      3.212E-01
   80.000         3.0401           0.0            0.06625        0.9337         4.185E-02      3.289E-01

  100.000         2.9321           0.0            0.05303        0.9470         4.339E-02      3.411E-01
  125.000         2.8384           0.0            0.04253        0.9575         4.482E-02      3.523E-01
  150.000         2.7695           0.0            0.03555        0.9644         4.594E-02      3.611E-01
  200.000         2.6754           0.0            0.02686        0.9731         4.755E-02      3.738E-01
  300.000         2.5701           0.0            0.01819        0.9818         4.950E-02      3.891E-01
  400.000         2.5117           0.0            0.01384        0.9862         5.065E-02      3.981E-01
  500.000         2.4742           0.0            0.01123        0.9888         5.142E-02      4.042E-01
  600.000         2.4478           0.00000        0.00946        0.9905         5.198E-02      4.085E-01
  700.000         2.4282           0.0            0.00820        0.9918         5.240E-02      4.118E-01
  800.000         2.4129           0.00000        0.00723        0.9928         5.273E-02      4.144E-01

 1000.000         2.3906           0.0            0.00589        0.9941         5.322E-02      4.183E-01
 1250.000         2.3719           0.00000        0.00480        0.9952         5.364E-02      4.216E-01
 1500.000         2.3588           0.00000        0.00406        0.9959         5.394E-02      4.239E-01
 2000.000         2.3415           0.0            0.00311        0.9969         5.434E-02      4.271E-01
 3000.000         2.3231           0.0            0.00215        0.9979         5.477E-02      4.305E-01
 4000.000         2.3131           0.0            0.00165        0.9983         5.500E-02      4.323E-01
 5000.000         2.3069           0.0            0.00135        0.9987         5.515E-02      4.335E-01
 6000.000         2.3026           0.0            0.00114        0.9989         5.525E-02      4.343E-01
 7000.000         2.2994           0.0            0.00099        0.9990         5.533E-02      4.349E-01
 8000.000         2.2970           0.0            0.00088        0.9991         5.539E-02      4.354E-01

10000.000         2.2934           0.0            0.00071        0.9993         5.548E-02      4.360E-01
12500.000         2.2905           0.00000        0.00058        0.9994         5.555E-02      4.366E-01
15000.000         2.2884          -0.00000        0.00050        0.9995         5.560E-02      4.370E-01
20000.000         2.2858           0.0            0.00038        0.9996         5.566E-02      4.375E-01
```

Figure 10: Output of examin.mortran (Photon part. Unix, PC and Macintosh). Line-width is shortened for a printing. Use examin.out file for diff command.

```
 1         SUMMARY OF ELECTRON DATA STORED FOR MEDIUM=FE-RAYLEIG
OELECTRON KINETIC ENERGY RANGE: 0.989   19999.488 MEV  PHOTON ENERGY RANGE:  0.100   20000.000 MEV
ORADIATION LENGTH=     1.76084 cm                                DENSITY=     7.8600 g/cm**3
0200.*TEFFO=      0.0594 cm

OKINETIC      STEP SIZE   e(-) DEDX  e(+) DEDX    ENERGY      e(-),e(+) MEAN FREE PATH(BREM FRACTION)
ENERGY MeV       cm         MeV/cm     MeV/cm     LOSS                          cm

    0.989     1.001E-02     10.368     10.113     0.105     9.173E-01( 1.00000)    6.797E-01( 0.74102)

    1.000     1.018E-02     10.363     10.104     0.105     9.114E-01( 1.00000)    6.761E-01( 0.74190)
    1.250     1.426E-02     10.311      9.971     0.118     8.065E-01( 1.00000)    6.129E-01( 0.75989)
    1.500     1.882E-02     10.325      9.920     0.130     7.398E-01( 1.00000)    5.724E-01( 0.77376)
    2.000     2.941E-02     10.416      9.919     0.153     6.497E-01( 0.99340)    5.193E-01( 0.79372)
    3.000     5.614E-02     10.340     10.021     0.193     4.802E-01( 0.86472)    4.542E-01( 0.81792)
    4.000     5.940E-02     10.375     10.133     0.154     4.174E-01( 0.84114)    4.129E-01( 0.83189)
    5.000     5.940E-02     10.426     10.228     0.124     3.819E-01( 0.83511)    3.845E-01( 0.84068)
    6.000     5.940E-02     10.475     10.308     0.104     3.563E-01( 0.83526)    3.615E-01( 0.84758)
    7.000     5.940E-02     10.519     10.374     0.089     3.377E-01( 0.83698)    3.440E-01( 0.85261)
    8.000     5.940E-02     10.558     10.430     0.078     3.237E-01( 0.83910)    3.304E-01( 0.85634)

   10.000     5.940E-02     10.624     10.520     0.063     3.040E-01( 0.84330)    3.105E-01( 0.86151)
   12.500     5.940E-02     10.687     10.603     0.051     2.855E-01( 0.84896)    2.915E-01( 0.86665)
   15.000     5.940E-02     10.737     10.666     0.043     2.722E-01( 0.85375)    2.775E-01( 0.87044)
   20.000     5.940E-02     10.809     10.756     0.032     2.536E-01( 0.86131)    2.579E-01( 0.87590)
   30.000     5.940E-02     10.897     10.861     0.022     2.322E-01( 0.87106)    2.353E-01( 0.88249)
   40.000     5.940E-02     10.948     10.921     0.016     2.194E-01( 0.87733)    2.218E-01( 0.88672)
   50.000     5.940E-02     10.982     10.960     0.013     2.106E-01( 0.88183)    2.125E-01( 0.88979)
   60.000     5.940E-02     11.005     10.987     0.011     2.039E-01( 0.88533)    2.055E-01( 0.89225)
   70.000     5.940E-02     11.023     11.007     0.009     1.986E-01( 0.88813)    2.000E-01( 0.89427)
   80.000     5.940E-02     11.037     11.023     0.008     1.943E-01( 0.89045)    1.955E-01( 0.89596)

  100.000     5.940E-02     11.056     11.044     0.007     1.875E-01( 0.89410)    1.884E-01( 0.89870)
  125.000     5.940E-02     11.070     11.062     0.005     1.812E-01( 0.89752)    1.820E-01( 0.90134)
  150.000     5.940E-02     11.080     11.073     0.004     1.764E-01( 0.90015)    1.770E-01( 0.90343)
  200.000     5.940E-02     11.091     11.085     0.003     1.694E-01( 0.90402)    1.699E-01( 0.90657)
  300.000     5.940E-02     11.098     11.095     0.002     1.605E-01( 0.90896)    1.609E-01( 0.91075)
  400.000     5.940E-02     11.100     11.097     0.002     1.548E-01( 0.91216)    1.551E-01( 0.91355)
  500.000     5.940E-02     11.100     11.098     0.001     1.507E-01( 0.91448)    1.509E-01( 0.91561)
  600.000     5.940E-02     11.100     11.098     0.001     1.475E-01( 0.91628)    1.476E-01( 0.91725)
  700.000     5.940E-02     11.100     11.099     0.001     1.449E-01( 0.91775)    1.450E-01( 0.91858)
  800.000     5.940E-02     11.100     11.099     0.001     1.427E-01( 0.91897)    1.428E-01( 0.91971)

 1000.000     5.940E-02     11.100     11.099     0.001     1.392E-01( 0.92094)    1.393E-01( 0.92154)
 1250.000     5.940E-02     11.100     11.099     0.001     1.359E-01( 0.92281)    1.360E-01( 0.92330)
 1500.000     5.940E-02     11.100     11.100     0.000     1.334E-01( 0.92427)    1.334E-01( 0.92469)
 2000.000     5.940E-02     11.100     11.100     0.000     1.295E-01( 0.92647)    1.295E-01( 0.92679)
 3000.000     5.940E-02     11.100     11.100     0.000     1.244E-01( 0.92935)    1.244E-01( 0.92957)
 4000.000     5.940E-02     11.100     11.100     0.000     1.210E-01( 0.93126)    1.211E-01( 0.93142)
 5000.000     5.940E-02     11.100     11.100     0.000     1.185E-01( 0.93266)    1.186E-01( 0.93280)
 6000.000     5.940E-02     11.100     11.100     0.000     1.166E-01( 0.93378)    1.166E-01( 0.93389)
 7000.000     5.940E-02     11.100     11.100     0.000     1.150E-01( 0.93469)    1.150E-01( 0.93478)
 8000.000     5.940E-02     11.100     11.100     0.000     1.136E-01( 0.93544)    1.136E-01( 0.93554)

10000.000     5.940E-02     11.100     11.100     0.000     1.114E-01( 0.93669)    1.115E-01( 0.93676)
12500.000     5.940E-02     11.100     11.100     0.000     1.093E-01( 0.93790)    1.093E-01( 0.93794)
15000.000     5.940E-02     11.100     11.100     0.000     1.077E-01( 0.93882)    1.077E-01( 0.93888)
19999.488     5.940E-02     11.100     11.100     0.000     1.052E-01( 0.94026)    1.052E-01( 0.94030)

OWHAT FRACTION SHOULD TMXS BE:
```

Figure 11: Output of examin.mortran (Electron part. Unix, PC and Macintosh). Line-width is shortened for a printing. Use examin.out file for diff command.