Geant4 Education

Python Interface to ENDL-EEDL/EPDL/EADL

FJPPL Workshop

@LAPP/Annecy

2008.09.03

Katsuya Amako

(KEK)

Aim of the project

- For a systematic study of Geant4 Physics processes, we want have a test framework. Koichi is now constructing it based on Python.
- Among many subjects to be studied, one of the most basic one is to validate cross sections of fundamental physics processes.
- We want to validate the current cross sections used in G4 electromagnetic processes by comparing available cross section databases.
- Decided to implement a Python interface to EEDL/EPDL/APDL database so that we can handly compare cross sections,

EEDL, EPDL and EADL

- EEDL: Evaluated Electron Data Library
- EPDL: Evaluated Photon Data Library
- EADL: Evaluated Atomic Data Library
- Available data rangel
  - Atomic number: 1 -100
  - Energy range: 10 eV - 100 GeV
- Database Format
  - All of them are based on
    ENDL - Evaluated Nuclear Database Library by LLNL
  - Reference:
    UCRL-ID-117796 Rev.1
  - File structure is simple though data format is not so simple

```
10000 0 0 2.01790+ 1 901205 2 0.00000+ 0 0.00000+00 0.00000+ 0   Header Part
91912 0 0.00000+ 0 0.00000+ 0 0.00000+ 0 0.00000+ 0 0.00000+ 0

1.00000+ 0 2.00000+ 0     Data Part
3.00000+ 0 2.00000+ 0
5.00000+ 0 2.00000+ 0
6.00000+ 0 4.00000+ 0
```

EEDL - Included Data

1) Elastic transport
   a) transport cross section, sigma (1-<cosq>) (b).
2) Large angle elastic scattering (over cosq = -l. to 0.999999)
   a) integrated large angle scattering cross section (b),
   b) average energy of the scattered electron (MeV),
   c) average energy to the residual atom, i.e., local deposition (MeV),
   d) angular distribution of the scattered electron.
3) Elastic scattering
   a) integrated scattering cross section (b).
4) Ionization (by subshell, over the recoil electron energy range down to 0.1 eV),
   a) integrated cross section (b),
   b) average energy of the scattered and recoil electron (MeV),
   c) spectra of the recoil electron (MeV-l).
5) Bremsstrahlung (over the photon energy range down to 0.1 eV),
   a) integrated cross section (b),
   b) average energy of the secondary electron and photon (MeV) ,
   c) spectra of the secondary photon (MeV-l).
6) Excitation
   a) integrated cross section (b),
   b) average energy to the residual atom, i.e., local deposition (MeV).

EPDL - Included Data

1) Coherent scattering,
   a) integrated cross section (b),
   b) form factor,
   c) real and imaginary anomalous scattering factors,
   d) average energy of the scattered photon (MeV),
2) Incoherent scattering
   a) integrated cross section (b),
   b) scattering function,
   c) average energy of the scattered photon and recoil electron (MeV).
3a) Total photoelectric reaction
   a) integrated cross section (b),
   b) average energy to the residual atom, i.e., local deposition (MeV),
   c) average energy of the secondary photons and electrons (MeV).
3b) Photoelectric reaction, by subshell
   a) integrated cross section (b),
   b) average energy to the residual atom, i.e., local deposition (MeV),
   c) average energy of the secondary photons and electrons (MeV) .
4) Pair production reaction
   a) integrated cross section (b),
   b) average energy of the secondary electron and positron (MeV) .
5) Triplet production reaction
   a) integrated cross section (b),
   b) average energy of the secondary electron and positron (MeV) .

EAPL - Included Data

1) Subshell data,
    a) number of electrons,
    b) binding and kinetic energy (MeV),
    c) average radius (cm),
    d) radiative and nonradiative level widths (MeV),
    e) average number of released electrons and x-rays,
    f) average energy of released electrons and x-rays (MeV),
    g) average energy to the residual atom, i.e., local deposition (MeV).
2) Transition probability data
    a) radiation transition probabilities,
    b) nonradiative transition probabilities.

Design of User Interface

```
>>> from EEDL_Database import *
        # Import EEDL_Database package
>>> dataBase = EEDL_Database("path to the original EEDL data file")
        # Create database from the original EEDL data file
>>> dataBase.ShowDatabaseSummary()
        # Show a summary of the created database
>>> dataBase.Help()
        # Show the help of database usage
>>> elementSymbol = "Li"
        # Want to retrieve the "Li" data
>>> dataTypeNum = 6
        # Specify data type number for retrieval - 6 means Elastic
        # scattering cross section. For specification of dataType
        #  --> see "List of dataType number for retrieving data" below
>>> tableHeader, tableData = dataBase.RetrieveData(elementSymbol, dataTypeNum)
        # Retrieve the data - the following two objects will be returned
        #  1. EEDL_Header object: contains header information in original EEDL file
        #  2. EEDL_Data   object: contains physics data, i.e. cross section, etc
        #  If the requested data is not available, two "None" objects will be returned
>>> dataTypeNum = 7
        # Retrieve ionization cross section which requires the following
        # subshell info
>>> subShell = 1
        # Specify K-shell - subShell value is a designator number defined
        # in "Atomic Subshell Designaors" in the EEDL document
        # If you failed to provide a valid subshell value, you will be a
        # retrieval error
>>> dataBase.RetrieveData(elementSymbol, dataTypeNum, subShell)
        # Retrieve the data - the following two objects will be returned
        #  1. EEDL_Header object: contains header information in original EEDL file
        #  2. EEDL_Data   object: contains physics data, i.e. cross section, etc
        #  If the requested data is not available, two "None" objects will be returned
```

Current Status

EEDL Ready to use

EPDL Next step

- not take a long time because I can reuse the same design

EADL Next to next step

- same as above