Sep. 27th

L07
Physics processes

Toshiyuki Toshito

Nagoya Proton Therapy Center

Nagoya City University West Medical Center

- Brief introduction of physics processes and physicslists
- Focused on the predefined PhysicsConstructors in Geant4 concerning to medical physics

Physics Lists:
Description of Physics Processes

Geant4 11.0 patch-03 準拠

Geant4講習会資料 : HEP/NP/Space 2022

KEK 大学共同利用機関法人
高エネルギー加速器研究機構

Contents

1. Essence of description of interactions between particles and materials
2. Reference Physics Lists
3. Practical usage of Physics Lists
4. Physics Lists for specific cases
5. Implementation of Modular Physics Lists

Essence of description of interactions between particles and materials

Physics of Geant4

- Geant4 is able to handle interactions of particles in materials based on accumulated knowledge of particle, nuclear and atomic physics
  - In this lecture, *Physics of Geant4* means knowledge of physics in traduced to Geant4 from above various fields
  - Physics of Geant4 forms the basis of Geant4 toolkit

- Results of simulation obtained by the users is derived from the Physics of Geant4
  - It is important for users to understand the basics of how “Physics of Geant4” is expressed in order to correctly interpret the results
  - In general, using simulation as a “black box” is dangerous

Physics processes and their classification

- Particles undergo various interactions as they move through matters
- Each interaction is called *Process* and described by a process class

  For example:

  - G4ComptonScattering ← photon Compton scattering
  - G4eBremsstrahlung ← e^+/e^- bremsstrahlung
  - ….

- Processes are classified as:
  - Electromagnetic
  - Hadronic
  - Decay
  - Parameterized
  - Transportation
  - Optical

Geant4 EM sub-libraries

- Standard
  - γ, e^+, e up to 100 TeV
  - hadrons up to 100 TeV
  - ions up to 100 TeV
- Muons
  - up to 1 PeV
  - energy loss propagator
- X-rays
  - X-ray and optical photon production processes
- High-energy
  - process at high energy (E>10GeV)
  - physics for exotic particles
- Polarization
  - simulation of polarized beams
- Optical
  - optical photon interactions

- Low-energy
  - Livermore library γ,e- from 10 eV up to 1 GeV
  - Livermore library based polarized processes
  - PENELOPE 2008 code rewrite γ,e-,e+ from 250 eV up to 6 GeV
  - hadrons and ions up to 1 GeV
  - atomic de-excitation

    (fluorescence + Auger)
- DNA
  - Geant4 DNA models and processes
  - Micro-dosimetry models for radiobiology from 0.025 eV to 10 MeV
  - many of them material specific (water)
  - Chemistry in liquid water

Geant4 Physics: Hadronic

- Pure hadronic interactions for 0 to 100 TeV
  - elastic, inelastic, capture, fission
- Radioactive decay:
  - both at-rest and in-flight
- Photo-nuclear interaction from ~1 MeV to 100 TeV
- Lepto-nuclear interaction from ~100 MeV up to 100 TeV
  - e- and e+ induced nuclear reactions
  - muon induced nuclear reactions

Physics models and processes

- Geant4 can handle physics processes of particles from low energy（below ‘eV’）to extremely high energy (above ‘PeV’) region
- However, there is no unified physics model applicable over all energy regions
- So, in general a number of physics models are available for one physics process in Geant4
  - For example, following models are provided for Compton scattering process :
    - G4KleinNishinaCompton
    - G4PenelopeComptonModel
    - G4LivermoreComptonModel
    - …….
  - Applicable energy region is limited for Each physics model
- Users need to determine the best combination of physics models and configure it for initialization
  - This information is called ***Physics List***, which is an object derived from G4VUserPhysicsList class

    🡐 next slide

Physics List

- *Physics List* is a class :
  - specify all the particles that will be used in the simulation application
  - together with the list of physics processes assigned to each individual particles
  - and production threshold of secondary particles (discussed later)

- *Physics List* is one out of the three mandatory classes that must be prepared by users, which requires adequate knowledge of Physics of Geant4
  - It is challenging for ordinary users
  - It is convenient and recommended to start with one of the *Reference Physics Lists*
    - Reference Physics Lists cover most of the “use cases” and are included in the toolkit
    - It is recommended that users select and use the one they think is most suitable from among these
    - Users can customizable (by calling appropriate methods before initialization)

☞ In the following lecture, we will give an overview of the physics models prepared by Geant4, and then explain the details of the *Reference Physics Lists*

EM Physics Constructors

G4EmStandardPhysics	for ATLAS and other HEP simulation applications
G4EmStandardPhysics_option1	similar to one used by CMS
G4EmStandardPhysics_option2	similar to one used by LHCb
G4EmStandardPhysics_option3	proton/ion therapy
G4EmStandardPhysics_option4	the goal is to have the most accurate EM physics description
G4EmLivermorePhysics	accurate Livermore based low energy e^- and γ transport
G4EmPenelopePhysics	accurate PENELOPE based low energy e^-,e^+ and γ transport
G4EmDNAPhysics	Geant4-DNA
G4OpticalPhysics	for optical photons
………	

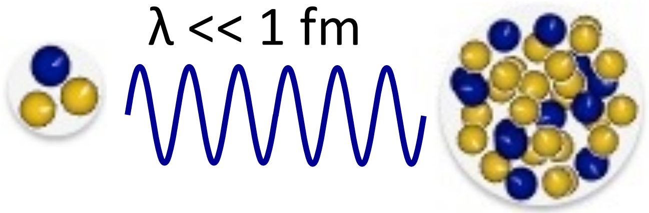
☞ For details, please refer to the following web site :

- http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/electromagnetic/index.html
- http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsReferenceManual/html/electromagnetic/index.html

Hadronic Interactions from TeV to meV

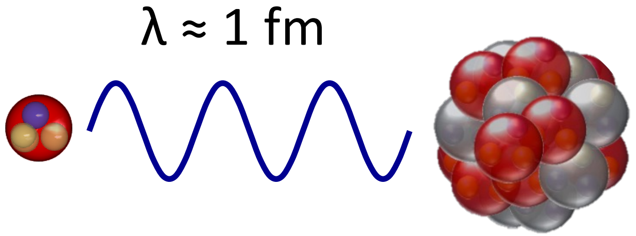
- High energy
  - Quark-gluon interaction
  - String model
    - Quark-Gluon string model
    - Fritiof

TeV hadron



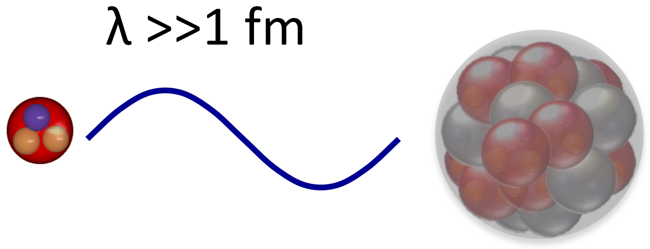
- Intermediate energy
  - Hadron-nucleon interaction
  - Intra-nuclear cascade model
    - Binary cascade model
    - Bertini model
    - INCL++

~GeV - ~100 MeV



- Low energy
  - Nucleon-nucleus interaction
  - Pre-compound model
    - Fission/Evaporation
    - Gamma Evaporation
    - Radioactive Decay
    - Capture at rest

~100 MeV - ~10 MeV

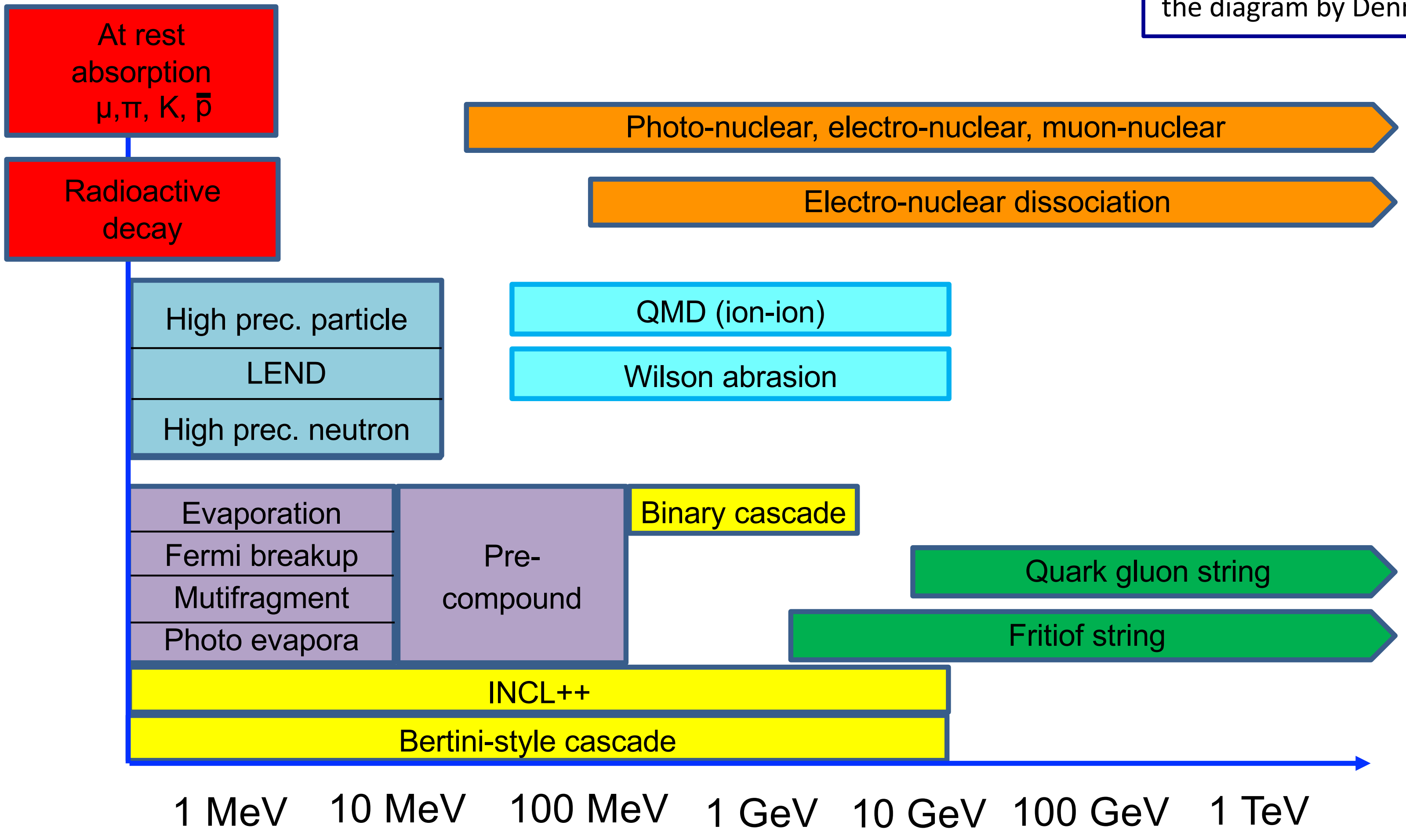


~10 MeV to thermal

The pictures courtesy by Marc Verderi.

Hadronic Processes

Updated version (2020) based on the diagram by Dennis Wright



☞ For details, please refer to the following web site :

- http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsReferenceManual/html/hadronic/index.html

Reference Physics Lists

What is Reference Physics Lists？

- *Reference Physics Lists* are pre-defined Physics Lists which respond to frequent “use cases”
  - When choosing a physics list: does it cover your needs ?
  - Of course, you are invited anyway to perform relevant validations
- You can find in *geant4/source/physics_lists/lists* of Geant4 source code
- Each *physics list* includes different choice of EM and hadronic physics
- These physics lists can be found on the Geant4 web page at
  - *http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/index.html*
- You can find validation information on the Physics of Geant4 on the following web page
  - *http://geant4.web.cern.ch/publications_validations/testing_and_validation*

Reference Physics Lists （Geant4 11.0,11.1）

related to FTF	related to QGS	Others
FTFP_BERT	QGSP_BERT	LBE
FTFP_BERT_ATL	QGSP_BERT_HP	NuBeam
FTFP_BERT_HP	QGSP_BIC	Shielding
FTFP_BERT_TRV	QGSP_BIC_AllHP	ShieldingLEND
FTFP_INCLXX	QGSP_BIC_HP	
FTFP_INCLXX_HP	QGSP_FTFP_BERT	
FTFQGSP_BERT	QGSP_INCLXX	
FTF_BIC	QGSP_INCLXX_HP	
	QGS_BIC	
	QBBC	

[Note]

- See the next slide for naming rules

- ■ 22 types of Reference Physics Lists are distributed
  - ● If you include the options provided in each List, total number will be much larger
- ■ For details on Reference Physics Lists, refer to the following web page

  http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/reference_PL/index.html

Naming rules for Physics Lists

- Hadron nuclear interaction models are specified with the following prefixes:
  - FTF -> Fritiof Parton String model : E > ~3 GeV
  - QGS -> Quark Gluon String model : E > ~12 GeV
  - BERT -> Bertini-style Cascade: E < ~ 10 GeV
  - BIC -> Binary Cascade: E < ~ 10 GeV
  - INCLXX -> Liege intra-nuclear cascade model (INCL) E < ~10 GeV
  - P -> Precompund model for nucear de-excitation E < ~150 MeV（for example, FTFP、QGSP, etc）
  - HP -> High Precision neutron model E < 20 MeV

[Note]
adaptive energy regions are rough indications

- Electromagnetic interaction models are specified with the following suffixes:
  - No suffix -> standard EM physics
  - _EMV, _EMX -> fast options for high-energy physics
  - _EMY, _EMZ , _LIV, _PEN -> more precise options, for medical and space science applications

- examples :

Name of list	EM	Low energy hadron	High energy hadron
FTFP_BERT	Standard model	Bertini model	Fritiof model
FTFP_BERT_LIV	Livermore model	Bertini model	Fritiof model
QGSP_BIC_PEN	Penelope model	Binary model	QGS model
....			

Recommended Reference Physics Lists

- Followings are stably maintained ones among many *Reference Physics Lists* :
  - FTFP_BERT
  - FTFP_BERT_HP
  - QGSP_BERT
  - QGSP_BERT_HP
  - QGSP_BIC
  - QGSP_FTFP_BERT
- It is recommended to start with above Reference Physics Lists
- FTFP_BERT : recommended by Geant4 for HEP
  - after the release of Geant4.10

☞ The following lecture will explain the basic characteristics of these Reference Physics Lists

Features of recommended Reference Physics Lists – 1

- *FTFP_BERT*
  - Standard electromagnetic physics
  - Uses Bertini-style cascade for hadrons < 5 GeV
  - Uses Fritiof model for high energies > 4 GeV
- *QGSP_BERT*
  - Similar to FTFP_BERT but using QGS (Quark Gluon String) model for high energies > 12 GeV
  - Fritiof model in between 9.5 – 25 GeV
- *QGSP_FTFP_BERT*
  - Similar to QGSP_BERT
  - Fritiof model in between 6 – 25 GeV

Features of recommended Reference Physics Lists － ２

- *FTFP_BERT_HP (QGSP_BERT_HP)*
  - Same as FTFP_BERT (QGSP_BERT) , but with the high-precision neutron model (NeutronHP) used for neutrons below 20 MeV
  - Significantly slower than FTFP_BERT (QGSP_BERT), especially when Doppler broadening on-the-fly is used
    - There is an option to turn this off
  - For radiation protection and shielding applications
- *QGSP_BIC*
  - Same as QGSP_BERT, but replaces Bertini-style cascade with Binary cascade model (+ Precompound model)
  - Recommended for use at energies below 200 MeV
    - Many medical applications
    - Suggested EM option: _EMY or _EMZ

Other Physics Lists

- *Shielding*
  - Based on FTFP_BERT_HP with improved neutron cross sections from JENDL
  - Better ion nuclear interactions using QMD model
  - Radioactive decay model activated
  - Currently used by SuperCDMS dark matter search
  - Recommended for: Shielding applications, Space physics, HEP
- *FTFP_INCLXX , FTFP_INCLXX_HP*
  - Like FTFP_BERT(_HP), but with Bertini-style cascade replaced by INCLXX (Liege) cascade model below 3 GeV
- *QGSP_BIC_HP*
  - Same as QGSP_BIC, but with the high precision neutron model (NeutronHP) used for neutrons below 20 MeV
  - Recommended for: Radiation protection, Medical applications
- *QBBC*
  - Used both Bertini-style and Binary cascade models
  - Latest coherent elastic scattering
  - Neutron XS approach (fairly accurate, but faster than HP)

Other Physics Lists (based on use-case) – 1

- If primary particle energy in your application is < 5 GeV (for example, clinical proton beam of 150MeV)
  - start with a physics list which includes “BIC” or ”BERT”
  - e.g. QGSP_**BIC**, QGSP_**BERT**, FTFP_**BERT**, etc.
- If neutron transport is important
  - start with a physics list containing “HP”
  - e.g. QGSP_BIC_**HP**, FTFP_BERT_**HP**, etc.
- If you are interested in Bragg curve physics
  - Use a physics list ending with “_EMV” or “EMX”
  - e.g. QGSP_BERT**_EMV**
- Ion – ion interactions
  - G4IonQMDPhysics
    - QMD model

Other Physics Lists (based on use-case) － ２

- Transportation of optical photons
  - Register G4OpticalPhysics in EM models with Physics List

- Radioactive Decay
  - Try Shielding first

- For detailed line Emissions from EM processes
  - EM options: **_EMY , _EMZ , _LIV , _PEN**

Practical usage of Physics Lists

Physics List object

- *Physics List* is a class which collects all
  - the particles,
  - physics process including a list of physics models
  - and production thresholds of secondary particles (Range Cuts) ⬅ see the next section

  needed for your application.

- *A Physics List object* is created using one of the following two types of classes
  1. G4VUserPhysicsList
     - It is one of the three mandatory classes that must exist in your simulation
     - The most fundamental class to create Physics List object
     - The most granular approach
  2. G4VModularPhysicsList
     - Derived from G4VUserPhysicsList
     - Classify physics processes information into categories using G4VPhysicsConstructor and register in G4VModularPhysicsList
       - Maintenance of Physics List is easier than G4VUserPhysicsList
       - Methods are provided that allows you to add and modify physics process information by category (module)
         - RegisterPhysics(G4VPhysicsConstructor*); to register module
         - RemovePhysics(G4VPhysicsConstructor*); to remove module
         - ReplacePhysics(G4VPhysicsConstructor*); to replace module

- All *Reference Physics Lists* are built with *G4VModularPhysicsList*
  - The physics lists handled by the users are created from G4VModularPhysicsList

How to use Physics List

- To make use of existing *physics lists*, you have two choices

  method #1

  Directly generate physics list object you want to use and pass it to *G4RunManager*

  - If you want to use the reference physics lists provided by Geant4 as is, this method is the easiest
  - When using a physics list created by the user, use this method
  - The physics list used is determined when the application is compiled

  method #2

  Pass the name of the reference physics list to *G4PhysListFactory*, get the physics list object, and then pass it to *G4RunManager*

  - You can add physics processes that are not included in the reference physics list provided by Geant4
  - You can change the EM model set in the reference physics list provided by Geant4

    ➡ The suffix for specifying the EM model mentioned in slide #18 can only be used with this method (see specific example later)
  - The physics list to be used can be specified at Run Time using environment variables

How to use Reference Physics Lists — method #1

method #1 : You can instantiate the physics list, and set it to the run manager

```
....
#include ”FTFP_BERT.hh”

int main( int argc,char** argv )
{
  // ....

  // Construct the default run manager
  G4RunManager * runManager = new G4RunManager{};

  // Physics list
  G4VModularPhysicsList* physicsList = new FTFP_BERT{};
  runManager->SetUserInitialization( physicsList );
  ....
  ....
```

include header file of reference physics list to use (FTFP_BERT)

create FTFP_BERT object

set it to Run Manager

[Note] More compactly

```
#include ”FTFP_BERT.hh”

int main( int argc,char** argv )
{
  // ....

  // Construct the default run manager
  G4RunManager * runManager = new G4RunManager{};

  // Physics list
  runManager->SetUserInitialization( new FTFP_BERT{} );
  ....
```

create FTFP_BERT object and immediately set it to Run Manager

How to use Reference Physics List ー method #2

method #2 : You can use *G4PhysListFactory* utility

```
....
#include "G4PhysListFactory.hh"

int main( int argc,char** argv )
{
 // ....
 // Physics List (choose a ready-made physics list)
   G4PhysListFactory  factory;
   G4VModularPhysicsList * physics =
    factory.GetReferencePhysList( "FTFP_BERT" );
   runManager->SetUserInitialization( physics );
....
....
```

include header file of G4PhyiscsFactory

generate G4PhyiscsFactory object

specify the name of physics list (“FTFP_BERT”) and get the physics list object you want to use from Factory

pass FTFP_BERT object to Run Manager

[Note 1] In the above example, the name of physics list is written directly in the code, but if you change the code as shown below, the name set in the environment variable "PHYSLIST" will be used

```
G4VModularPhysicsList* physicsList = factory.ReferencePhysList(“”);
```

[Note 2] To change the EM model set in the reference physics list, do the following

```
G4VModularPhysicsList* physicsList = factory.ReferencePhysList(“FTFP_BERT_XXX”);
                                                  // where XXX = EMV, EMX, LIV, PEN, etc
```

Physics Lists and threshold for secondary production (Range Cuts)

Range Cuts

- What’s a range cut ?
  - Secondary production threshold distance
  - Needed to secure CPU performance of simulation
  - Cuts in range are defined for ; e^- , e^+ , γ, p
  - Particles are tracked down to zero kinetic energy
  - For ionization, the cut secondary particles release all their energy at the point of generation, ensuring energy conservation

  [Note]

  ➢ Only the generated secondary particles are subject to range cutting

  ➢ If you want to cut primary particles, use G4UserLimits class

- Implicit and explicit usage of range cuts
  - implicit usage :
    - When implementing physical processes that have a cross section that diverges in the far infrared (bremsstrahlung radiation, ionization (δ-rays), and elastic scattering, etc.), an energy cut is required to avoid divergence, and Geant4 applies the above range cut value to this.

      ← Range cuts are automatically applied for these physics processes
  - explicit usage :
    - Range cuts are not automatically applied to secondary particles that have undergone physical processes that have no implicit usage (Compton scattering, gamma conversion etc.)

      ← Users need to explicitly command to apply range cuts(see next slide)

Configuration of Range Cuts

- Range Cuts settings when using Physics List
  - By default, Range Cuts is defined globally
  - Geant4 proposes the default value of 0.7 mm

- How to change default cut value
  - The default cuts values are set in the SetCuts() method of G4VUserPhysicsList class
  - They can be defined with UI command, eg:

```
Idle> /run/setCut 2.0 mm
```

[Note] User commands must be executed in the ’Idle’ state

  - It is recommended to use a range cut smallest dimension you’re interested in
  - It is possible to have different cut in range for particle type

```
Idle> /run/setCutForAGivenParticle e- 5.0 mm
```

[Note] Setting different cut values for each particle will affect energy loss calculations

  - It is possible to define specific cut in range per G4Region

- Points to note when changing range cut values
  - In some cases, particles below the cut value may not be cut
    - ex) Positrons are not cut because they have an at-rest process
  - Cut for proton is used for all hadrons and ions by elastic scattering processes
    - It is a cut on recoil ion kinetic energy

- Explicit usage of range cuts ⬅ See previous slide

```
Idle> /run/particle/applyCuts true gamma
```

Implementation of Modular Physics Lists

If the users want to create their own Physics List
without using References Physics Lists

Advanced topic

Modular Physics List (modularized Physics List)

- Physics Lists are described so as *G4VUserPhysicsList* to be their base-class
- However, a realistic ”Physics List” has many particles and physics processes
  - Writing such a physics list as a “flat list” makes it long,
  - Complicated, hard to read,
  - And hard to maintain !
- Recommendation
  - Use *G4VModularPhysicsList* (called Modular Physics List) as a base
  - This class is defined by inheriting from *G4VUserPhysicsList* and is convenient for users
- Geant4 adopted a “modular physics list” approach using G4VModularPhysicsList class
  - Physics is organized by “physics modules”

    ex) EM_Physics, Hadron_Physics,…
  - And you register the physics modules you are interested in in your “modular physics list”

G4VModularPhysicsList and G4VPhysicsConstructor

- The “physics modules” class registered in G4VModularPhyiscsList is:

  G4VPhysicsConstructor

- A physics constructor holds information about a physical interaction and all the particles involved in that interaction
- If you create an object for the physics constructor, interactions will be set for all particles
- All physics lists provided by Geant4 use this physics constructor
  - You can find them in *geant4/source/physics_lists/constructors*
  - They are classified into the following categories

    decay/ electromagnetic/ gamma_lepto_nuclear/
    hadron_elastic/ hadron_inelastic/ ions/
    limiters/ stopping/

- When users create their own physics list, we recommend that they first select what they need from these physics constructors and combine them with G4VModularPhysicsList
  - See the next slide for specific examples

Implementation of Physics List using physics constructor

- Example code based on Geant4 *Basic Example B3*

header file

```
//+++++++++++++++++++++++++++++++++++++++++++++++++++
// MyPhysicsList.hh
//+++++++++++++++++++++++++++++++++++++++++++++++++++
#ifndef MyPhysicsList_h
#define MyPhysicsList_h 1

#include "G4VModularPhysicsList.hh"

//---------------------------------------------------
  class MyPhysicsList : public G4VModularPhysicsList
//---------------------------------------------------
{
  public:
    MyPhysicsList();
   ~MyPhysicsList();

  public:
    void SetCuts();
};
#endif
```

MyPhysicsList is-a modular physics list

source file

```
//++++++++++++++++++++++++++++++++++++++++++++++++++++++
// MyPhysicsList.cc
//   [Note] Based on "G4 Basic Example: B3"
//++++++++++++++++++++++++++++++++++++++++++++++++++++++
#include ”MyPhysicsList.hh"
#include "G4DecayPhysics.hh"
#include "G4RadioactiveDecayPhysics.hh"
#include "G4EmStandardPhysics.hh"

//------------------------------------------------------
  MyPhysicsList::MyPhysicsList()
  : G4VModularPhysicsList()
//------------------------------------------------------
{
// Default physics
   RegisterPhysics(new G4DecayPhysics{});

// Radioactive decay
   RegisterPhysics(new G4RadioactiveDecayPhysics{});

// EM physics
   RegisterPhysics(new G4EmStandardPhysics{});
}

//------------------------------------------------------
  MyPhysicsList::~MyPhysicsList()
//------------------------------------------------------
{}

//------------------------------------------------------
  void MyPhysicsList::SetCuts()
//------------------------------------------------------
{
    G4VUserPhysicsList::SetCuts();
}
```

Three registered physics processes are provided by Geant4 as a physics constructor

See the next slide for the implementation of SetCuts()

Implementation of SetCuts()

- The values of *Range cuts* are set in the SetCuts() method

- In the example on the previous slide, the SetCuts value is set to the value implemented in G4VUserPhysicsList, which is the base class of G4VModularPhysicsList.

- Another example of implementation of SetCuts:

```
void PhysicsList::SetCuts()
{
  defaultCutValue = 1.0*mm;

  SetCutValue(0.5*mm, “e-”);
  SetCutValue(0.5*mm, “e+”);
}
```

to define ‘defaultCutValue’

to define cut values different from ‘defaultCutValue’ for specific particles

How to describe physics constructor

- In the previous example, we created a Physics List using the physics constructor provided by Geant4
- It you want to use your own physics constructor, define a class as below.

```
Class MyPhysConstructor : public G4VPhysicsConstructor
{
  public:
    MyPhysConstructor (const G4String& name = “MyPhysConstructor”);
    virtual ~MyConstructor();
    virtual void ConstructParticle();
    virtual void ConstructProcess();
}
```

← define physics constructor
（inherit G4VPhysicsConstructor）

- Implement two methods of above user-defined class
  - ConstructParticle() method ← to declare the particle types used
  - ConstructProcess() method ← to associate processes to above particles
  - Refer to following codes for practical example of these methods
    - Geant4/physics_lists/constructors/decay/include/G4DecayPhysics.hh
    - Geant4/physics_lists/constructors/decay/src/G4DecayPhysics.cc