



Geant4 @ NCBJ, 2022

7. Fizyka

Autor Autor

Przemysław Adrich



NARODOWE
CENTRUM
BADAŃ
JĄDROWYCH
ŚWIERK



Plan kursu

Wykłady

1. Wstęp do Geant4. Rys historyczny. Zastosowania. Przegląd możliwości. Instalacja. Dokumentacja.
2. Podstawowa struktura kodu. Hierarchie klas. Klasy użytkownika (obowiązkowe, opcjonalne).
~~Interfejsy. System jednostek. Liczby losowe. Śledzenie przebiegu symulacji („verbosity”).~~
3. Geometria i materiały.
4. Detektory typu „primitive scorer”, „probe”.
5. Detektory użytkownika. Histogramy i n-tuple.
6. Obiekty typu „UserAction” jako detektory. (Phasespace).
7. **Fizyka. Źródło. Wizualizacja.**
8. Niepewność statystyczna w obliczeniach Monte Carlo. Geant4 na klastrze CiŚ.

Przegląd zagadnień pozostawionych na przyszłość:

- wielowątkowość („multithreading”),
- własne interfejsy („messengers”),
- interfejs Roota (histogramy, n-tuple), interfejs python
- redukcja wariancji, „physics biasing”, „event biasing”, „geometrical biasing”,
- fotony optyczne, fizyka hadronowa, procesy i cząstki użytkownika,
- obcięcia energetyczne zależne od cząstki, regionu geometrii,
- zmiany geometrii i detektorów w trakcie wykonania programu,
- pole EM,
- światy równoległe,
- trackInformation, eventInformation, runInformation
- „stacking”,
- fast simulation,
- import geometrii z CAD,
- periodic boundary conditions,
- specjalistyczne kody bazujące na Geant4 (G4Beamline, GAMOS, GATE ...),
- ...

* „Monte Carlo First Run”
(wykład bonusowy)

Wykład w (bardzo) dużej mierze bazuje na:

- G. Folger, „*Physics Lists*”, 7 October 2021, G4 Advanced Tutorial @ CERN
- A. Ribon, „*Hadronic Physics I*”, 4 – 8 October 2021, G4 Advanced Tutorial @ CERN
- D. Wright, „*Hadronic Physics I*”, 7 February 2019, Geant4 Tutorial at Sao Paulo
- V. Ivantchenko, „*Electromagnetic Physics*”, 4-8 October 2021, G4 Advanced Tutorial @ CERN

[!\[\]\(4222901cb0b23d3b20e48e0aa550c263_img.jpg\) Geant4 Homepage](#)

Physics Reference Manual

The Physics Reference Manual provides detailed explanations of the physics implemented in the GEANT4 toolkit.

The manual's purpose is threefold:

- to present the theoretical formulation, model, or parameterization of the physics interactions included in GEANT4,
- to describe the probability of the occurrence of an interaction and the sampling mechanisms required to simulate it, and
- to serve as a reference for toolkit users and developers who wish to consult the underlying physics of an interaction.

Physics Reference Manual

Scope of this Manual

- 458 stron (dostępne jako książka w [pdf](#), [epub](#), [mobi](#))
- Ponad 300 referencji odnośnie samej fizyki EM i ponad 200 odnośnie fizyki hadronowej.

[!\[\]\(b29945fa2a6be561a89902fe3bc1eee0_img.jpg\) Geant4 Homepage](#)

PhysicsListGuide

This guide is a description of the physics lists class which is one of the mandatory user classes for a GEANT4 application. For the most part the "reference" physic lists included in the source distribution are described here as well the modularity and electronic options. Some use cases and areas of application are also described.

Guide for Physics Lists

Scope of this Manual

```
#include "G4RunManager.hh"
#include "DetectorConstruction.hh"
#include "ActionInitialization.hh"
#include "PhysicsList.hh"

int main(int argc,char** argv)
{
    G4RunManager* runManager = new G4RunManager;
    runManager->SetUserInitialization(new DetectorConstruction());
    runManager->SetUserInitialization(PhysicsList); ←
// User action initialization
    runManager->SetUserInitialization(new ActionInitialization());
...
}
```

PhysicsList musi być wskaźnikiem do obiektu klasy pochodnej od G4VUserPhysicsList.

Uwaga! Fizyka musi być zainicjalizowana i zarejestrowana do RunManagera przed inicjalizowaniem UserSteppingAction!

Fizyka – czym jest PhysicsList?

- **Physics List jest obiektem odpowiedzialnym za:**
 - określenie jakiego rodzaju cząstki będą mogły pojawić się w symulacji,
 - określenie jakim procesom fizycznym będą mogły podlegać cząstki poszczególnych rodzajów.
- **Nie ma domyślnej PhysicsList – musi być zawsze świadomie zbudowana przez użytkownika.**
- **Jeden z 3 obowiązkowych obiektów, które muszą zostać stworzone przez użytkownika i zarejestrowane do RunManagera.**
 - Jest dla RunManagera jedynym źródłem informacji o tym kiedy, w jaki sposób i które procesy fizyczne powinny być wywoływane.
- **Zapewnia bardzo elastyczny sposób „konfigurowania praw fizyki”:**
 - użytkownik może wybrać cząstki, które powinny być używane,
 - użytkownik może wybrać procesy fizyczne, którym podlegać będą cząstki.
- **ALE, użytkownik musi być dobrze (a nawet bardzo dobrze) zorientowany, jaką „część fizyki” jest niezbędna do poprawnego opisania problemu:**
 - pominięcie istotnych cząstek i/lub oddziaływań nie spowoduje błędu wykonania programu, ale wartość uzyskanych wyników będzie marna,
 - uwzględnienie zbędnych cząstek i procesów może znakomicie wydłużyć czas uzyskania wyników bez żadnego wpływu na ich jakość (np. neutrony termiczne w eksperymencie fizyki wysokich energii).

Fizyka – jak zbudować Physics List?

Poziom trudności

1. Na bazie klasy bazowej [G4VUserPhysicsList](#)

- Zapewnia tylko podstawowy interfejs
- Trzeba „ręcznie” wyspecyfikować wszystkie cząstki jakie będą używane
- Dla każdej cząstki trzeba „ręcznie” określić jakim procesom ma podlegać
- Dla procesów hadronowych oprócz określenia modeli trzeba wybrać źródła przekrojów czynnych
- Sporo kodu do napisania ...
- Trudne do utrzymania i wspierania. Obecnie niezalecane.

albo

2. Na bazie [G4VModularPhysicsList](#)

- Wzbogacony i rozwinięty interfejs, **dużo łatwiejsze w użyciu**
- Dostępny szeroki wachlarz gotowych tzw. „physics constructors”, które w dużej mierze eliminują potrzebę ręcznego określania cząstek i procesów.
- Można rozbudować o własne „konstruktory” (pochodne od [G4VPhysicsConstructor](#)).

albo

3. Wykorzystać jeden z kilku wzorcowych zestawów „Physics List”

- Wywołując bezpośrednio lub za pomocą „fabryczki”: [G4PhysListFactory](#)
- Łatwo dostosować i zmodyfikować pod konkretne potrzeby

Warygodność. Wsparcie.

G4VUserPhysicsList

Fizyka - G4UserPhysicsList

- Podstawowy interfejs listy fizyki w Geant4 (klasa abstrakcyjna).
- Każda, niezależnie którą metodą zbudowana, lista fizyki musi być pochodną (dziedziczyć po G4UserPhysicsList).
- Historycznie najstarsza metoda specyfikowania fizyki.
- Wymaga największej (dogłębnej) wiedzy/świadomości stosowanej fizyki.
- Użytkownik **musi** zaimplementować dwie czysto wirtualne metody:
 - **ConstructParticle()** – jawne utworzenie wszystkich cząstek potrzebnych w symulacji, także tych, które mogą pojawić się jako cząstki wtórne,
 - **ConstructProcess()** – jawne utworzenie procesów i przypisanie procesów do cząstek.
- Opcjonalnie **można** zaimplementować metodę **SetCuts()** – do ustawienia progów produkcji cząstek wtórnych

```
4  class YourPhysicsList: public G4VUserPhysicsList {  
5      public:  
6          // CTR  
7          YourPhysicsList();  
8          // DTR  
9          virtual ~YourPhysicsList();  
10  
11         // pure virtual => needs to be implemented  
12         virtual void ConstructParticle();  
13         // pure virtual => needs to be implemented  
14         virtual void ConstructProcess();  
15  
16         // virtual method  
17         virtual void SetCuts();  
18         ...  
19         ...  
20     };
```

Fizyka - G4UserPhysicsList::ConstructParticle()

Cząstki można konstruować indywidualnie, jedna po drugiej wywołując ich konstruktory:

- w G4 wybór jest szeroki
<https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/particle.html#particle-list-in-geant4>
- w tle automatycznie tworzona jest G4ParticleTable – obiekt przechowujący informację o wszystkich typach cząstek dostępnych w danej aplikacji.

```
void YourPhysicsList::ConstructParticle() {  
    G4Electron::Definition();  
    G4Gamma::Definition();  
    G4Proton::Definition();  
    G4Neutron::Definition();  
    // other particle definitions  
    ...  
    ...  
}
```

Alternatywnie cząstki można konstruować grupami, wg. ich rodzajów (wywołując odpowiednie konstruktory):

- Leptony
- Bariony
- Mezony
- Jony
- ...

```
void YourPhysicsList::ConstructParticle() {  
    // construct baryons  
    G4BaryonConstructor baryonConstructor;  
    baryonConstructor.ConstructParticle();  
    // construct bosons  
    G4BosonConstructor bosonConstructor;  
    bosonConstructor.ConstructParticle();  
    // more particle definitions  
    ...  
    ...  
}
```

Fizyka - G4UserPhysicsList::ConstructParticle()



User Support

1. [Getting started](#)
2. [Training courses and materials](#)
3. [Source code](#)
 - a. [Download page](#)
 - b. [LXR code browser](#) (circled in red)
 - c. [doxygen documentation](#) (circled in red)
 - d. [GitHub](#)
 - e. [GitLab @ CERN](#)

A screenshot of the Geant4 v11.0.1 documentation website. The header includes "Geant4 v11.0.1" and a navigation bar with "Main Page", "Namespaces", "Classes", "Files" (highlighted with a red circle), and "Examples". Below the header, it says "Geant4 Documentation" and has a "File List" button.

geant4-v11.0.1
source
analysis
digits_hits
error_propagation
event
externals
g3tog4
geometry
global
graphics_reps
intercoms
interfaces
materials
parameterisations
particles
adjoint
bosons
hadrons
leptons
include
G4AntiNeutrinoE.hh
G4AntiNeutrinoMu.hh
G4AntiNeutrinoTau.hh
G4Electron.hh
G4LeptonConstructor.hh
G4MuonMinus.hh
G4MuonPlus.hh
G4NeutrinoE.hh
G4NeutrinoMu.hh
G4NeutrinoTau.hh
G4Positron.hh

Fizyka - [G4UserPhysicsList::ConstructProcess\(\)](#)

Proces w G4

- opisuje prawdopodobieństwo reakcji (całkowity przekrój czynny)
- modeluje wynik reakcji, tj. stan końcowy

Metoda ConstructProcess()

- Musi jawnie wywołać metodę AddTransportation() (metoda klasy [G4UserPhysicsList](#)) odpowiedzialną za utworzenie procesu transportu cząstek.
- Dla wygody może być podzielona na wywołania pomocniczych metod np. do osobnego utworzenia procesów elektromagnetycznych, hadronowych, itp.

```
void YourPhysicsList::ConstructProcess() {
    // method (provided by the G4UserPhysicsList base class)
    // that assigns transportation process to all particles
    // defined in ConstructParticle()
    AddTransportation();
    // helper method might be defined by the user (for convenience)
    // to add electromagnetic physics processes
    ConstructEM();
    // helper method might be defined by the user
    // to add all other physics processes
    ConstructGeneral();
}
```

Fizyka - G4UserPhysicsList::ConstructProcess()

```
void YourPhysicsList::ConstructEM() {  
    // get the physics list helper  
    // it will be used to assign processes to particles  
    G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();  
    auto particleIterator = GetParticleIterator(); ←  
    particleIterator->reset();  
    // iterate over the list of particles constructed in ConstructParticle()  
    while( (*particleIterator)() ) {  
        // get the current particle definition  
        G4ParticleDefinition* particleDef = particleIterator->value();  
        // if the current particle is the appropriate one => add EM processes  
        if ( particleDef == G4Gamma::Definition() ) {  
            // add physics processes to gamma particle here  
            ph->RegisterProcess(new G4GammaConversion(), particleDef); ←  
            ...  
            ...  
        } else if ( particleDef == G4Electron::Definition() ) {  
            // add physics processes to electron here  
            ph->RegisterProcess(new G4eBremsstrahlung(), particleDef);  
            ...  
            ...  
        } else if (...) {  
            // do the same for all other particles like e+, mu+, mu-, etc.  
            ...  
        }  
    }  
}
```

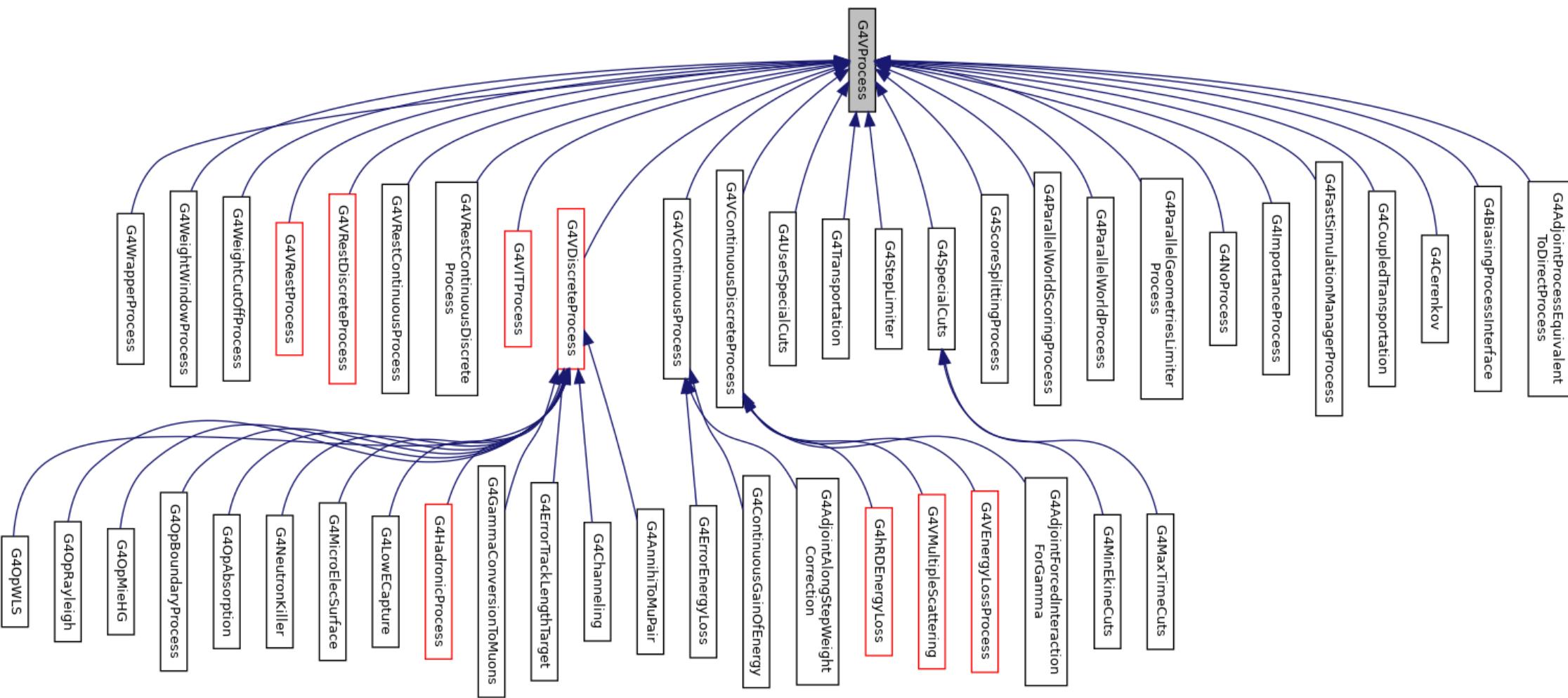
Zastrzeżona metoda klasy bazowej.

Pojawiła się stosunkowo niedawno.

G4PhysicsListHelper dba o właściwe uszeregowanie procesów tak by, np. procesy AlongStep były wywoływanie przed procesami PostStep. Dzięki temu nie musimy tego robić sami.

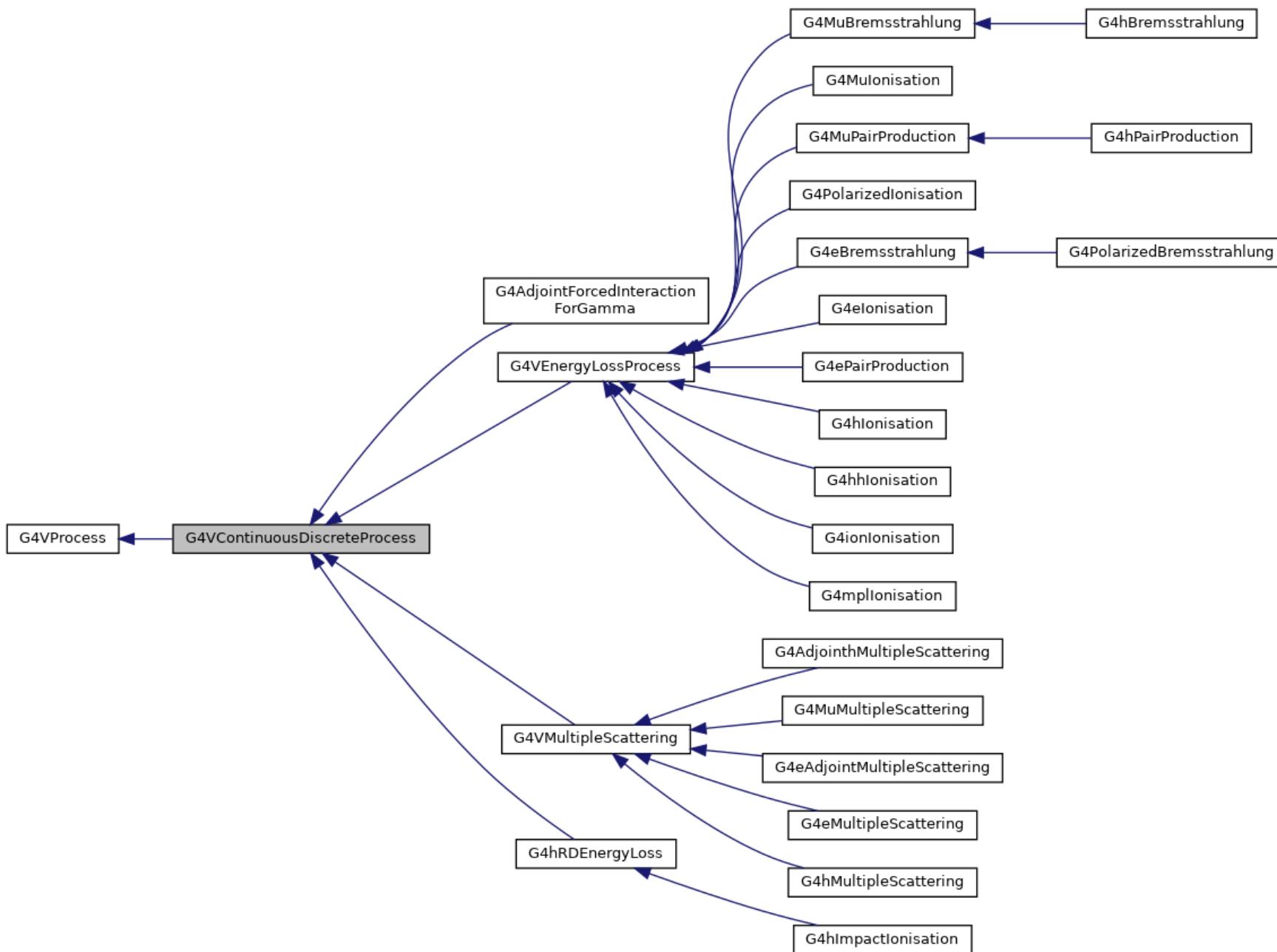
Fizyka – procesy – klasa bazowa G4VProcess

<https://geant4.kek.jp/Reference/v11.0.1/classG4VProcess.html>

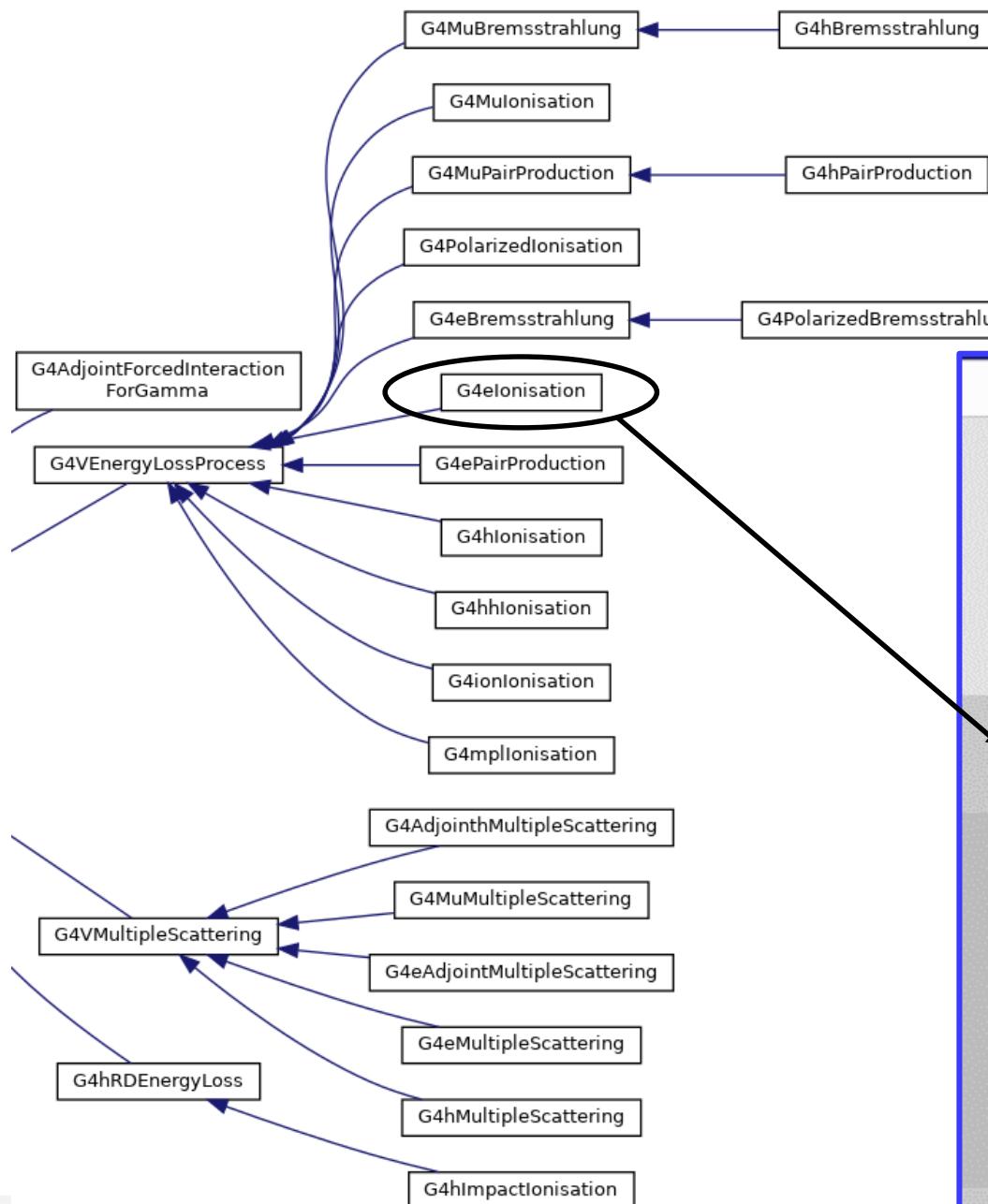


Fizyka – procesy – klasy pochodne dla różnych klas procesów

<https://geant4.kek.jp/Reference/v11.0.1/classG4VProcess.html>

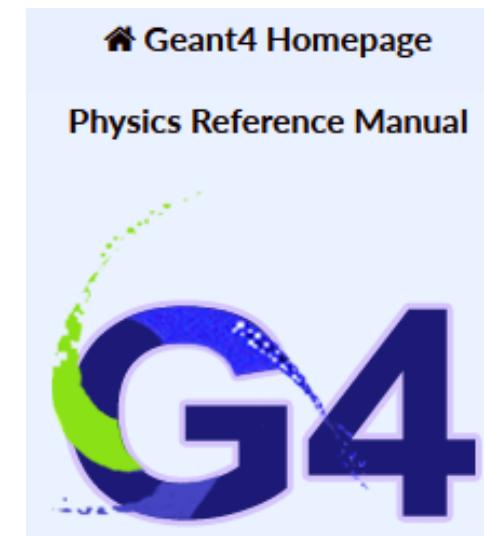


Fizyka – procesy – klasy pochodne dla różnych klas procesów



Szczegółowa dokumentacja w „**Physics Reference Manual**”

- Np. ponad 300 referencji odnośnie samej fizyki EM i ponad 200 odnośnie fizyki hadronowej.
- 458 stron (dostępne jako książka w [pdf](#), epub, mobi)



G4VModularPhysicsList

Fizyka - G4VModularPhysicsList

G4VModularPhysicsList nadbudowuje i rozszerza **G4VUserPhysicsList** o wiele przydatnych funkcjonalności. (*Subiektywna opinia: naprawdę świetny wynalazek!*)

- Znacznie wygodniejszy (i bezpieczniejszy) sposób tworzenia PhysicsList w oparciu o tzw. moduły,
- Moduł reprezentuje całą grupę (kategorię) procesów i cząstek,
- Dostępnych jest wiele modułów: np. procesy elektromagnetyczne, procesy rozpraszania elastycznego hadronów, oddziaływania jonów, ...
- Proces transportu jest zawsze dodawany automatycznie,
- Moduły są reprezentowane przez obiekty klas pochodnych od [G4VPhysicsConstructor](#).
Nazewnictwo wprowadza trochę zamieszania, bo często moduły określane są „konstruktorami fizyki” co jest raczej mało jednoznaczne.
- G4VModularPhysicsList udostępnia kilka wygodnych metod do operacji na modułach, np.:
 - RegisterPhysics(G4VPhysicsConstructor*)
 - GetPhysics(...), poprzez indeks, nazwę lub typ
 - ReplacePhysics(G4VPhysicsConstructor*)
 - RemovePhysics(...), poprzez indeks, nazwę lub typ
- Można tworzyć własne moduły na bazie G4VPhysicsConstructor.

Uwaga na marginesie.

G4VModularPhysicsList nie jest klasą abstrakcyjną.

Da się utworzyć obiekt tej klasy i na nim wykonywać operacje typu RegisterPhysics(). Można go też bezpośrednio zarejestrować do RunManagera:

runManager->SetUserInitialization(G4VModularPhysicsList).*

Czyli w skrócie, aby korzystać z modułów, nie trzeba koniecznie tworzyć własnej klasy pochodnej od G4VModularPhysicsList, choć jest to zalecane ze względu na czytelność i serwisowalność kodu.

Fizyka – Typy modułów (konstruktorów fizyki)

- Moduł tworzy pewien podzbiór procesów fizycznych (wraz z niezbędnymi częściami).
 - *Konkretny proces fizyczny może być utworzony wyłącznie jednokrotnie.*
- Każdy moduł ma swój specyficzny typ, który jest używany do:
 - zapewnienia, że najwyżej jeden moduł danego typu znajduje się na „liście fizyki” (PhysicsList)
 - pobrania, zmiany, usunięcia modułu z PhysicsList
- Istniejące typy modułów (zdefiniowane w G4BuilderType.hh):
 - bUnknown
 - bTransportation
 - bElectromagnetic
 - bEmExtra
 - bDecay
 - bHadronElastic
 - bHadronInelastic
 - bStopping
 - bIons

Fizyka – Moduły (konstruktory fizyki)

- Moduły są pogrupowane w kategorie: Electromagnetic, hadron_elastic, hadron_inelastic, gamma_lepto_nuclear, ions, decay, stopping, limiters
- Pełna lista modułów (konstruktorów): https://geant4.kek.jp/lxr/source/physics_lists/constructors/
- Niektóre standardowe konstruktory/moduły EM (w sumie jest ich ponad 30):
 - **G4EmStandardPhysics** – domyślny (eksperyment ATLAS i inne HEP)
 - **G4EmStandardPhysics_option1** – dedykowany dla HEP (eksperyment CMS), szybki ale mało dokładny
 - **G4EmStandardPhysics_option2** – dedykowany dla HEP (eksperyment LHCb), eksperimentalny
 - **G4EmStandardPhysics_option3** – dedykowany (i zalecany) dla fizyki medycznej i zastosowań w badaniach kosmicznych. Wyższa dokładność kosztem dłuższego czasu wykonania.
 - **G4EmStandardPhysics_option4** – zestaw najbardziej dokładnych modeli EM (najdłuższy czas wykonania)
- Niektóre standardowe konstruktory/moduły hadronowe:
 - **G4HadronElasticPhysics** – domyślne modele rozpraszania elastycznego na jądrach. Dla wszystkich hadronów.
 - **G4HadronElasticPhysicsHP** – jw. ale z użyciem modeli wysokiej precyzji (HP) dla neutronów poniżej 20 MeV.
 - **G4HadronPhysicsFTFP_BERT** – rozpraszanie nieelastyczne na jądrach. Dla wszystkich hadronów.
 - **G4IonPhysics** – oddziaływanie jonów.
- Zalecana lektura „*Physics List Guide*” i „*Physics Reference Manual*” oraz/lub wykładów z cyklu „Geant4 Advanced Course” nt. fizyki EM, hadronowej, fotonów optycznych, rozpadów cząstek nietrwałych, ...

Fizyka – Przykładowy szkic klasy pochodnej od G4VModularPhysicsList()

```
class YourModularPhysicsList : public G4VModularPhysicsList {
public:
    // CTR
    YourModularPhysicsList();
    ...
};

// CTR implementation
YourModularPhysicsList::YourModularPhysicsList()
: G4VModularPhysicsList() {
    // set default cut value (optional)
    defaultCutValue = 0.7*CLHEP::mm;
    // use pre-defined physics constructors
    // e.g. register standard EM physics using the pre-defined constructor
    // (includes constructions of all EM processes as well as the
    // corresponding particles)
    RegisterPhysics( new G4EmStandardPhysics() );
    // user might create their own constructor and register it
    // e.g. all physics processes having to do with protons (see below)
    RegisterPhysics( new YourProtonPhysics() );
    // add more constructors to complete the physics
    ...
}
```

Wzorcowe (referencyjne) Physics Lists

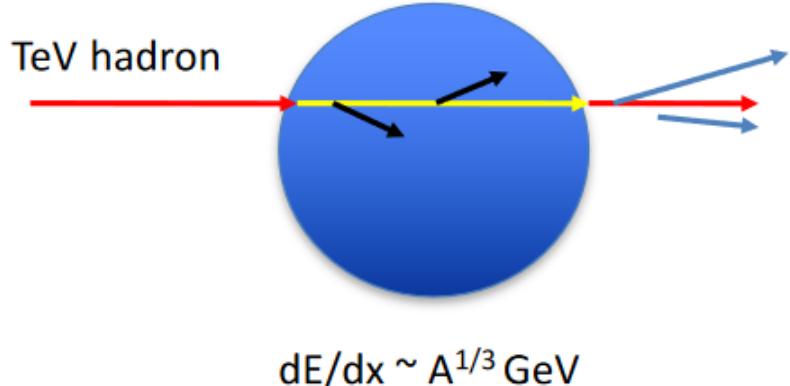
Fizyka – wzorcowe (referencyjne) listy fizyki

Geant4 zapewnia dużą liczbę wzorcowych (referencyjnych) list fizyki

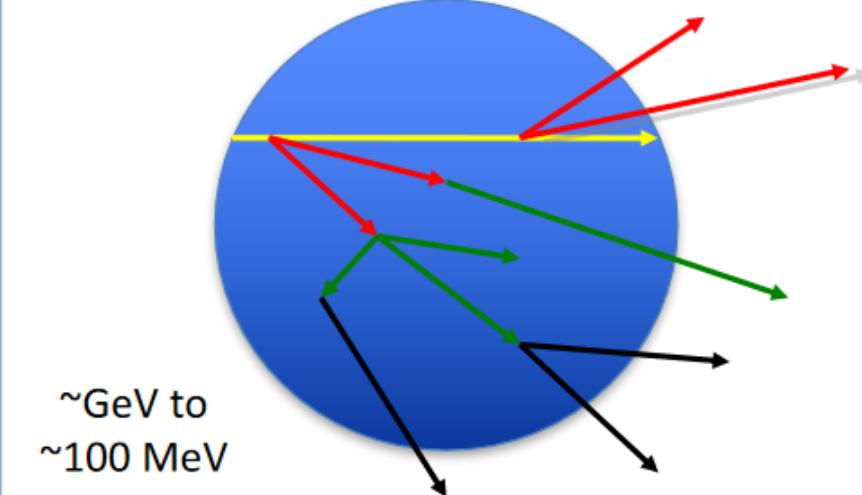
- „gotowe do użycia”, kompletne listy ukierunkowane na różne przypadki użycia,
- tworzone i utrzymywane przez ekspertów, często we współpracy z użytkownikami,
- ale nikt nie zagwarantuje, że dana lista jest „poprawna” lub optymalna dla danego przypadku.
- Użytkownik jest odpowiedzialny za sprawdzenie czy wybrana lista fizyki nadaje się do jego zastosowania.
- Pierwotnie stworzone, aby pomóc użytkownikom budować listy fizyki z uwzględnieniem fizyki hadronów
 - np. w nieelastycznym procesie hadronowym kilka, co najmniej dwa, różne modele muszą być połączone,
 - żaden model hadronowy w Geant4 nie obejmuje pełnego zakresu energii,
 - wybór modeli do połączenia wymaga wiedzy eksperckiej i wyników validacji,
 - modele często mają mocne i mniej mocne strony ⇒ trzeba oceniać i wybierać

Hadronic Interactions from TeV to meV

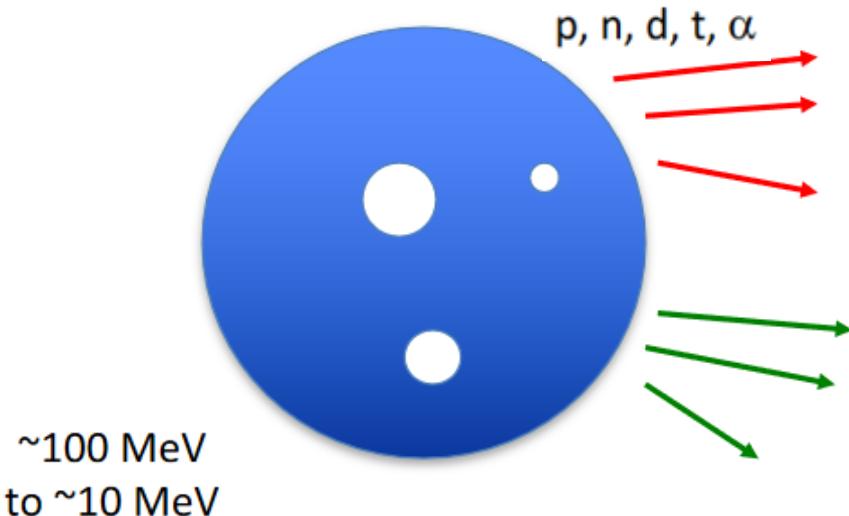
String model



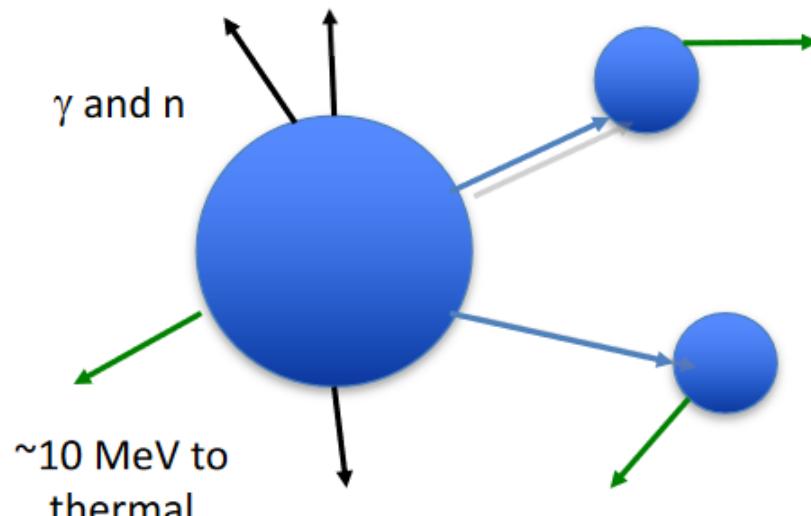
Intra-nuclear cascade model



Pre-equilibrium (Precompound) model



Equilibrium (Evaporation) model



Fizyka – rzut oka na fragment krajobrazu fizyki hadronowej

At rest
absorption, μ ,
 π , K, anti-p

Photo-nuclear, electro-nuclear

Radioactive
decay

Electro-nuclear dissociation

High precision
neutron

QMD (ion-ion)

Wilson Abrasion

Evaporation
Fermi breakup
Multifragment
Photon Evap

Pre-
compound

Quark Gluon string

Binary cascade

Fritiof string

BERT Intranuclear cascade

1 MeV 10 MeV 100 MeV 1 GeV 10 GeV 100 GeV 1 TeV

Fizyka – konwencja nazewnictwa list referencyjnych

- W większości przypadków nazwa listy referencyjnej odzwierciedla nazwę konstruktora (modułu) hadronowej fizyki nieelastycznej, opcjonalnie z dołączoną nazwą wybranego modułu fizyki EM.
 - Część hadronowa składa się z członów wskazujących na modele zastosowane kolejno od najwyższych do najniższych energii:
 - powyżej kilku (dziesiątek) GeV: „high energy string model: **QGS** or **FTF**”. Dodanie w nazwie litery „**P**” czyli **QGSP** lub **FTFP** oznacza, że dołączony jest moduł obsługujący deekscytację jądra wzburzonego („**Precompound + evaporation models**”)
 - Poniżej ~10 GeV: „intra nuclear cascade model” **BERT**, **BIC**, **INCLXX**
 - Poniżej 20 MeV: „**HP**” (high precision). Np. neutrony do energii termicznych (rozpr. elastyczne, nieelastyczne, absorpcja, rozszczepienie). Opcjonalnie. Bardzo wolne.
 - Możliwe wyjątki w postaci różnych skrótów opisujących specjalne warianty: np. **TRV**, **LEND**
 - Nazwy opcji fizyki elektromagnetycznej:
 - **EMV** = EM_option1
 - **EMX** = EM_option2
 - **EMY** = EM_option3
 - **EMZ** = EM_option4
 - **DNA**, **GS**, **Liv**, **Pen**, **LE**, **WVI**, **SS** – warianty do specjalnych zastosowań
- Istnieją specjalizowane referencyjne listy fizyki, których nazwy są wyjątkami od powyższego schematu: **Shielding**, **LBE** oraz **NuBeam**

FTFP_BERT

Aktualnie rekomendowana lista fizyki dla HEP (fizyki wysokich energii)

Główne składniki:

- **FTF** (Fritiof) hadronic string model, used above 3 GeV
 - **BERT** (Bertini-like) intra-nuclear cascade model, used below 6 GeV
 - Nucleus de-excitation: **Precompound + evaporation models**
 - Neutron capture
 - Nuclear capture of negatively charged hadrons at rest
 - Hadron elastic
 - Gamma- , electron- , and muon-nuclear
 - Standard electromagnetic physics
 - Bez uwzględnienia: neutron-HP , radioactive decay, optical photons
- Pomiędzy 3 a 6 GeV jeden z modeli wybierany jest losowo

Fizyka – nazewnictwo list referencyjnych – więcej przykładów

- **FTFP_BERT_HP** : as FTFP_BERT, but with **NeutronHP** for neutrons $E_{kin} < 20$ MeV
 - **Shielding** : similar to FTFP_BERT_HP, but using for ions also the **QMD** (Quantum Molecular Dynamics) model
 - QMD used in the range [100 MeV, 10 GeV] : below BIC, above FTFP
- **FTFP_INCLXX** : similar to FTFP_BERT, but using **INCLXX** for some particles
 - Protons, neutrons, charged pions below 20 GeV; FTFP above 15 GeV
- **QGSP_BERT** : similar to FTFP_BERT, but using **QGS** (Quark Gluon String) model at high energies
 - [3, 6] GeV transition BERT – FTFP ; [12, 25] GeV transition FTFP – QGSP
- **QGSP_BIC** : similar to FTFP_BERT but using QGS and BIC (Binary Cascade) instead of FTF and BERT when possible
 - Protons, neutrons : BIC < 6 GeV , FTFP in [3, 25] GeV , QGSP > 12 GeV
 - Pions & kaons : BERT < 6 GeV , FTFP in [3, 25] GeV , QGSP > 12 GeV

Fizyka – referencyjne listy fizyki – jak tego używać?

- Potencjalnie list fizyki do utrzymania byłoby ogromnie dużo:
 - Wszystkie możliwe kombinacje hadronowe × wszystkie opcje EM
- Rozwiązaniem zastosowanym w Geant4 jest dostarczenie podzbioru możliwych list hadronowych ze standardową fizyką EM – są to tzw. **listy referencyjne**
- Listy referencyjne są utrzymywane i walidowane przez twórców Geanta.
- Na bazie list referencyjnych, za pomocą „fabryczki” **G4PhysListFactory** można łatwo budować listy z alternatywnymi modelami fizyki EM (EMV, EMX, EMY, EMZ) i/lub dodawać inne moduły (np. rozpad promieniotwórczy)

```
// create a physics list factory object that knows
// everything about the available reference physics lists
// and can replace their default EM option
G4PhysListFactory physListFactory;
// obtain the QGSP_BIC_HP_EMZ reference physics lists
// which is the QGSP_BIC_HP refrence list with opt4 EM
const G4String plName = "QGSP_BIC_HP_EMZ";
G4VModularPhysicsList* pList = physListFactory.GetReferencePhysList(plName);
// (check that pList is not nullptr, that I skipp now)
// register your physics list in the run manager
runManager->SetUserInitialization(pList);
// register further mandatory objects i.e. Detector and Primary-generator
...
```

FTFP_BERT
FTFP_BERT_ATL
FTFP_BERT_HP
FTFP_BERT_TRV
FTFP_INCLXX
FTFQGSP_BERT
FTF_BIC
QBBC
QGSP_BERT
QGSP_BERT_HP
QGSP_BIC
QGSP_BIC_AIHP
QGSP_BIC_HP
QGSP_FTFP_BERT
QGSP_INCLXX
QGS_BIC
Shielding
ShieldingLEND
LBE
NuBeam

Fizyka – szczególne listy referencyjne, tzw. „Production physics lists”

- Kilka referencyjnych list fizyki, na których dokumentację, utrzymywanie i walidowanie kładzie się szczególnie duży nacisk
- Stosowane przez duże grupy użytkowników (eksperymenty przy LHC, fizyka medyczna)
- Wiarygodność jest głównym celem. Zmiany są robione rzadko i zachowawczo.
- Aktualnie listami „produkcyjnymi” są:

FTFP_BERT	the current G4 default, used in HEP collider experiments
QBBC	space physics and medical applications
QGSP_BERT	An early G4 default, was used by LHC experiments
QGSP_BIC	medical/hadrontherapy, normally used with option3 or option4 electromagnetic physics
Shielding	deep shielding applications, uses HP low energy neutron transport

- Szczegółowa dokumentacja w „**Physics List Guide**”
<http://cern.ch/geant4-userdoc/UsersGuides/PhysicsListGuide/html/index.html>

Fizyka – Modyfikowanie listy fizyki

- Referencyjne listy fizyki są obiektami typu G4VModularPhysicsList
- Można do nich dodawać moduły za pomocą konstruktorów fizyki i metod klasy G4VModularPhysicsList
 - np. pList->RegisterPhysics(new G4RadioactiveDecayPhysics)
- G4PhysListFactory udostępnia też kilka poleceń interfejsu użytkownika (menu /physics_lists/factory) do modyfikowania listy referencyjnej, np. addRadioactiveDecay, addOptical

Fizyka – Jak wybrać/zbudować fizykę?

- **Przypadek idealny:** użytkownik dobrze rozumie fizykę odpowiednią dla danej aplikacji
 - może zdecydować się na skorzystanie z predefiniowanej listy fizyki lub zbudować własną,
 - wybrana lista fizyki wymaga walidacji dla danej aplikacji,
 - może to zrobić użytkownik lub ktoś inny w przypadku niektórych list referencyjnych,
 - podczas procedury walidacji niektóre części listy fizyki mogą zostać zmienione.
- **Przypadek mniej idealny:** dana aplikacja należy do dobrze zdefiniowanego obszaru (np. medyczna)
 - użytkownik może wybrać listę referencyjną fizyki zalecaną dla danego obszaru aplikacji jako punkt wyjścia,
 - wybrana lista fizyki wymaga walidacji dla danej aplikacji (tak samo jak powyżej).
- **Przypadek jeszcze mniej idealny / beznadziejny:**
 - Szukamy czegoś możliwie zbliżonego wśród oficjalnych przykładów ([examples/extended/](#), [examples/extended/{electromagnetic, hadronic}](#), [examples/extended/physicslists/](#), [/examples/extended/optical/](#), [/examples/extended/radioactivedecay/](#), [/examples/extended/polarisation/](#), [/examples/extended/exoticphysics/](#), [/examples/advanced/](#)).
 - Szukamy porady u ekspertów (np. forum dyskusyjne G4).
- W przypadkach **jeszcze mniej idealnych** może też czasami zadziałać następująca procedura:
 - wybieramy najdokładniejsze ustawienia fizyki (np. opt4 dla EM). Niestety w przypadku fizyki hadronowej generalnie nie jest to możliwe.
 - uruchamiamy symulację – aby mieć punkt odniesienia,
 - krok po kroku rewidujemy początkową listę fizyki, używając dokładnych wyników jako odniesienia.
 - Do ostatecznych symulacji możemy świadomie wybrać mniej dokładne, ale szybsze ustawienie fizyki (np. opt0 dla EM)

<https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4PerformanceTips>

Selection of Physics Lists

Geant4 physics can be constructed by user from components, however, this requires some expertise. More simple solution is to use one of Reference Physics Lists provided in Geant4 Physics List library. Different approaches and variants of Physics Lists are shown in Geant4 novice, extended and advanced examples. The choice depends on concrete task and required accuracy:

- If only electromagnetic physics is needed for an application the best choice would be to use Physics Lists from extended electromagnetic examples ([TestEm3](#));
- For hadron therapy and similar applications also hadronic processes should be taken into account. The way how to add hadronic physics is shown in several extended examples ([TestEm7](#));
- If main physics processes in an application are hadronic then it is possible to use reference Physics Lists via *G4PhysicsListFactory* class, as it is shown in extended hadronic example ([Hadr00](#));

<https://twiki.cern.ch/twiki/bin/view/Geant4/Geant4PerformanceTips>

Selection of options for Geant4 electromagnetic (EM) physics

The main parameter of Geant4 EM simulation is the production cut. A user defines cut in range. The default value in reference Physics Lists is 0.7 mm. At initialisation stage of Geant4 values of production thresholds in energy for electrons and gamma are computed per material used in the simulation. For lower cuts number of simulated secondary particles increases and correspondingly increases CPU time. The choice of the cut value depends on the concrete task and geometry:

- Qualitatively the value of the cut corresponds to accuracy of simulation of energy deposition in space, so cut in range should be about the smallest thickness of absorber layers;
- If in the setup there are very different geometry parts which require very different accuracy (for example, micro-detector inside a complex apparatus), then the setup can be subdivided by different geometry regions and range cuts can be defined per region.

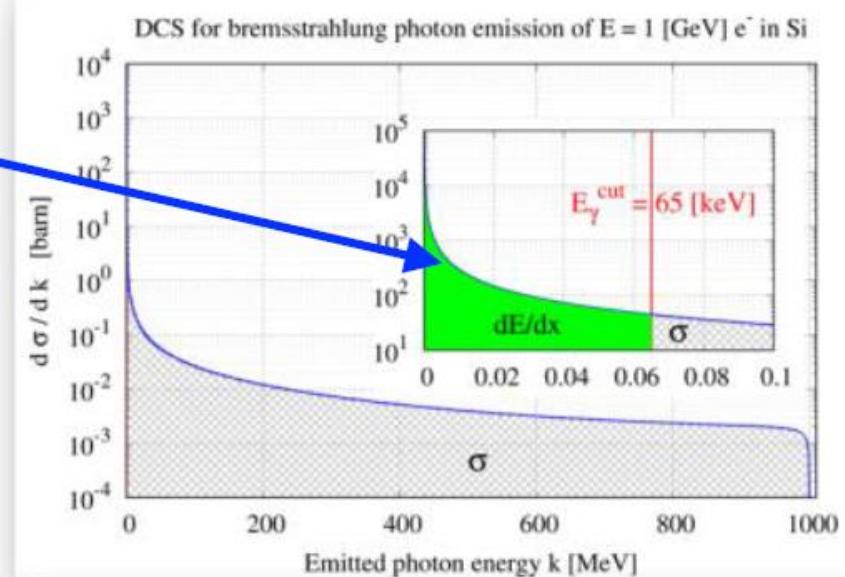
There are other options for EM physics. It is possible to choose different level of accuracy by selecting different constructors of the EM part of Physics Lists. Few guidelines to create an efficient EM physics:

- Try to use if possible [constructors](#) provided with Geant4 source code or examples;
- Start validation of a new application using the standard EM physics;
- EM standard models usually provides similar or better accuracy and much better CPU performance for particles with energy above 1 MeV;
- For electrons and gamma below 1 MeV some models of low-energy EM physics are more precise (for example, angular distribution of bremsstrahlung, Doppler broadening of Compton scattering);
- For muons, hadrons and ions standard EM physics should be used;
- If X-ray emission or Auger electron production are required then use low-energy Livermore EM physics;
- For simulation of PIXE low-energy Livermore EM physics should be used;
- For more details about accuracy and performance of EM physics, please, visit EM validation [page](#).

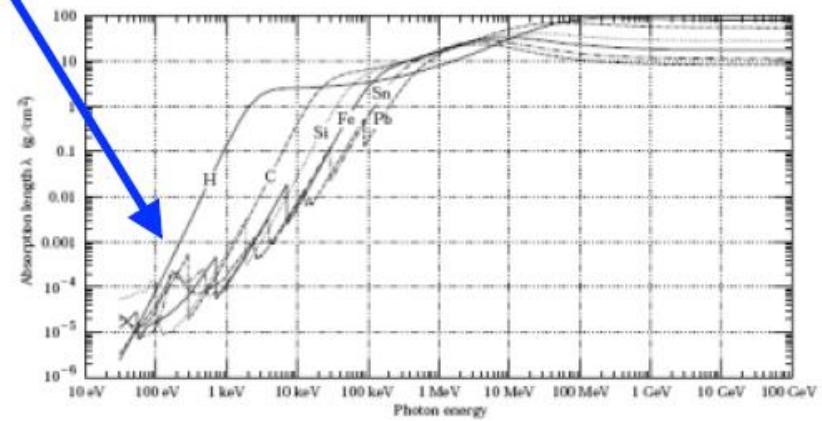
-- JohnApostolakis - First version: 04 Dec 2008, Last Edits February 2009

Fizyka – Progi produkcji

- Bremsstrahlung photon emission:
 - low energy photons (k small) will be emitted with high rate i.e. DCS $\sim 1/k$
 - generation and tracking of all these low energy photons would not be feasible (CPU time)
 - but low energy photons has a very small absorption length
 - If the detector spacial resolution is worst than this length (i.e. all volume boundaries are further), then the followings are equivalent:
 - a: generating and tracking these low energy photons till all their energy will be deposited
 - b: or just depositing the corresponding energy at the creation point (i.e. at a trajectory point)
 - note, that we think in energy scale at the model level that translates to length (spacial) at the transport level
 - a secondary production threshold might be introduced (either in energy or length)

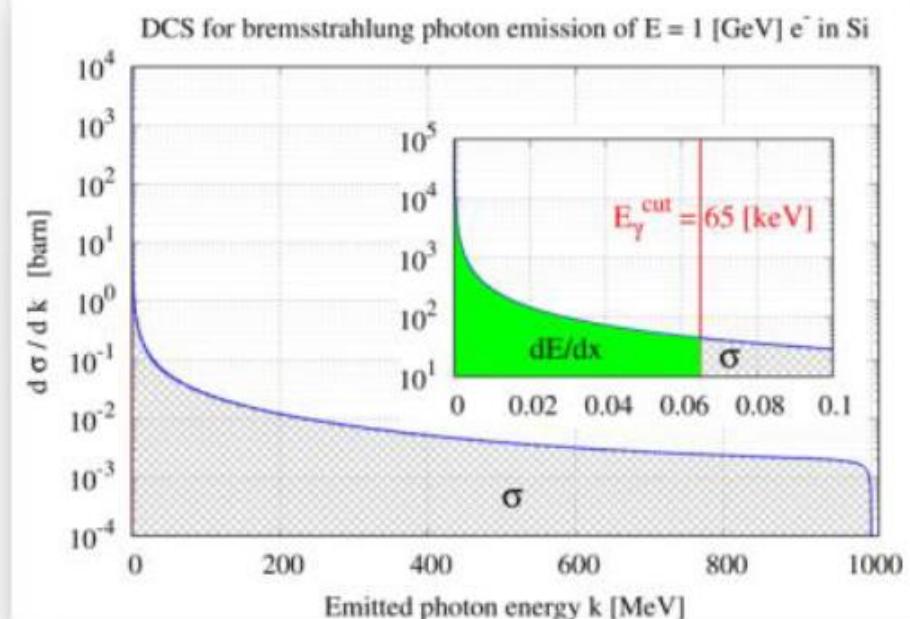


22.27. Passage of particles through matter



Fizyka – Progi produkcji

- Introduce secondary photon production threshold:
 - secondary photons, with initial energy below a gamma production threshold($k < E_{\gamma}^{\text{cut}}$), are not generated
 - the corresponding energy (that would have been taken away from the primary) is accounted as **CONTINUOUS** energy loss of the primary particle along its trajectory



- Electron makes a step with a given length L , one can compute the mean energy loss (due to sub-threshold photon emissions) along the step as $L \times dE/dx$ (would be true only if $E = \text{const}$ along the step)

$$\frac{dE}{dx}(E, E_{\gamma}^{\text{cut}}, Z) = N \int_0^{E_{\gamma}^{\text{cut}}} k \frac{d\sigma}{dk}(E, Z) dk$$

- Secondary photons, with initial energy above a gamma production threshold ($k > E_{\gamma}^{\text{cut}}$), are generated (**DISCRETE**)
- the emission rate is determined by the corresponding (restricted) cross section(σ)

$$\sigma(E, E_{\gamma}^{\text{cut}}, Z) = \int_{E_{\gamma}^{\text{cut}}}^E \frac{d\sigma}{dk}(E, Z) dk$$

Fizyka – Progi produkcji

- Klasa bazowa **G4VUserPhysicsList** udostępnia kilka metod do ustawiania wartości progów produkcji:

```
void SetDefaultCutValue(G4double newCutValue);
G4double GetDefaultCutValue() const;
    // Set/get the default cut value. Calling SetDefaultCutValue() causes
    // re-calculation of cut values and physics tables just before the
    // next event loop.

void SetCutValue(G4double aCut, const G4String& pname);
    // Sets a cut value for a particle type for the default region.

G4double GetCutValue(const G4String& pname) const;
    // Gets a cut value for a particle type for the default region.

void SetCutValue(G4double aCut, const G4String& pname,
                 const G4String& rname);
    // Sets a cut value for a particle type for a region.

void SetParticleCuts(G4double cut, G4ParticleDefinition* particle,
                     G4Region* region = nullptr);
void SetParticleCuts(G4double cut, const G4String& particleName,
                     G4Region* region = nullptr);
    // Invoke SetCuts for specified particle for a region.
    // If the pointer to the region is NULL, the default region is used
    // In case of "Retrieve" flag is ON, cut values will be retrieved
    // from files.

void SetCutsForRegion(G4double aCut, const G4String& rname);
    // Invoke SetCuts() for all particles in a region.
```

https://geant4.kek.jp/Reference/v11.0.1/G4VUserPhysicsList_8hh_source.html

Fizyka – Progi produkcji

- Progi produkcji cząstek wtórnych podajemy jako zasięg (domyślnie w mm).
- Domyślnie ustawione na 1 mm (0,7 mm w przypadku referencyjnych list fizyki).
- Optymalna wysokość progu jest zależna od aplikacji (rozmiary objętości aktywnych; CPU) i trzeba ją dobrać samemu.
- Wewnętrznie, w trakcie inicjalizacji, Geant4 przelicza zasięg na energię (uwzględniając materiał i rodzaj cząstki). Tu można podejrzeć jak to jest robione.
- Odpowiadająca progowi energia też ma domyślne minimum: 1 keV.
Jeśli podamy zasięg odpowiadający niższej energii to nie będzie to miało żadnego efektu, o ile jednocześnie nie zmieni się ustawienia poleceniem: /cuts/setLowEdge (np. /cuts/setLowEdge 500 eV)
- Progi produkcji są zdefiniowane tylko dla następujących cząstek wtórnych: **gamma, e⁻, e⁺, proton**
- Domyślnie progi zdefiniowane dla tych cząstek są stosowane, odpowiednio:
 - **gamma** – w procesie wytwarzania promieniowania hamowania (Bremsstrahlung)
 - **e⁻** – w procesie jonizacji
 - **e⁺** – produkcja pary e⁻e⁺
 - **proton** – próg stosowany jest jako minimalna energia odrzutu jądra w procesach elastycznych **wszystkich hadronów i jonów**.
- Opcjonalnie (jeśli ustawi się /process/em/applyCuts true) progi dla **gamma** i **e⁻** mogą być stosowane we wszystkich procesach dyskretnych EM wytwarzających te cząstki wtórne – rozpraszanie Comptona, zjawisko fotoelektryczne, ...
- Nie ma obowiązku stosować progów produkcji. Aczkolwiek wiele symulacji było by bez nich niewykonalnych.

- **Obowiązek użytkownika:** sprawdzenie (walidacja) czy wybrana/stworzona lista fizyki jest poprawna i odpowiednia do danego zagadnienia:
 - porównanie z pomiarami testowymi / obliczeniami analitycznymi (jeśli możliwe),
 - porównanie z danymi literaturowymi,
 - w ostateczności porównanie z wynikami z innego, wiarygodnego kodu MC.
- Przy przenoszeniu starego kodu do nowej wersji Geanta fizyka powinna być ponownie sprawdzona.
- Subiektywna opinia wykładowcy: *bez walidacji fizyki symulacja MC jest stratą czasu.*

- Twórcy Geanta walidują zaimplementowane modele fizyki (np. porównując wyniki testowych symulacji z wynikami pomiarów).
- Walidacja modeli jest procesem ciągłym, powtarzanym przy każdym nowym wydaniu.
- Procedury i wyniki walidacji są dokumentowane tutaj:
https://geant4.web.cern.ch/publications_validations/testing_and_validation

Fizyka - Podsumowanie

- Wszystkie cząstki, procesy fizyczne i progi produkcji potrzebne w symulacji muszą być zdefiniowane za pośrednictwem jednego obiektu: listy fizyki
- Dostępne są dwa rodzaje interfejsu listy fizyki:
 - **G4VUserPhysicsList** – w praktyce stosowalne tylko w najprostszych przypadkach
 - **G4VModularPhysicsList** – umożliwia budowę fizyki z gotowych modułów (konstruktorów)
- Wzorcowe(referencejne)/produkcyjne listy fizyki są udostępniane przez twórców Geanta.
 - Do ich wywołania i konfiguracji części elektromagnetycznej służy „fabryczka” **G4PhysListFactory**.
 - Listy wzorcowe pokrywają dość szeroki zakres różnych typowych zastosowań.
 - Składniki list wzorcowych można łatwo podmieniać (moduły)
- Dobranie odpowiedniej fizyki do danego zastosowania wymaga namysłu i może nie być proste.
- Wybrana **lista fizyki musi być walidowana** przed rozpoczęciem właściwych symulacji.

Dziękuję za uwagę



NARODOWE
CENTRUM
BADAŃ
JĄDROWYCH
ŚWIERK

www.ncbj.gov.pl

