



Geant4 @ NCBJ, 2022

8. Źródło

Autor Autor

Przemysław Adrich



**NARODOWE
CENTRUM
BADAŃ
JĄDROWYCH
ŚWIERK**



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. **Źródło**
9. Niepewność statystyczna w obliczeniach Monte Carlo. Geant4 na klastrze CiŚ.

Przegląd zagadnień pozostawionych na przyszłość:

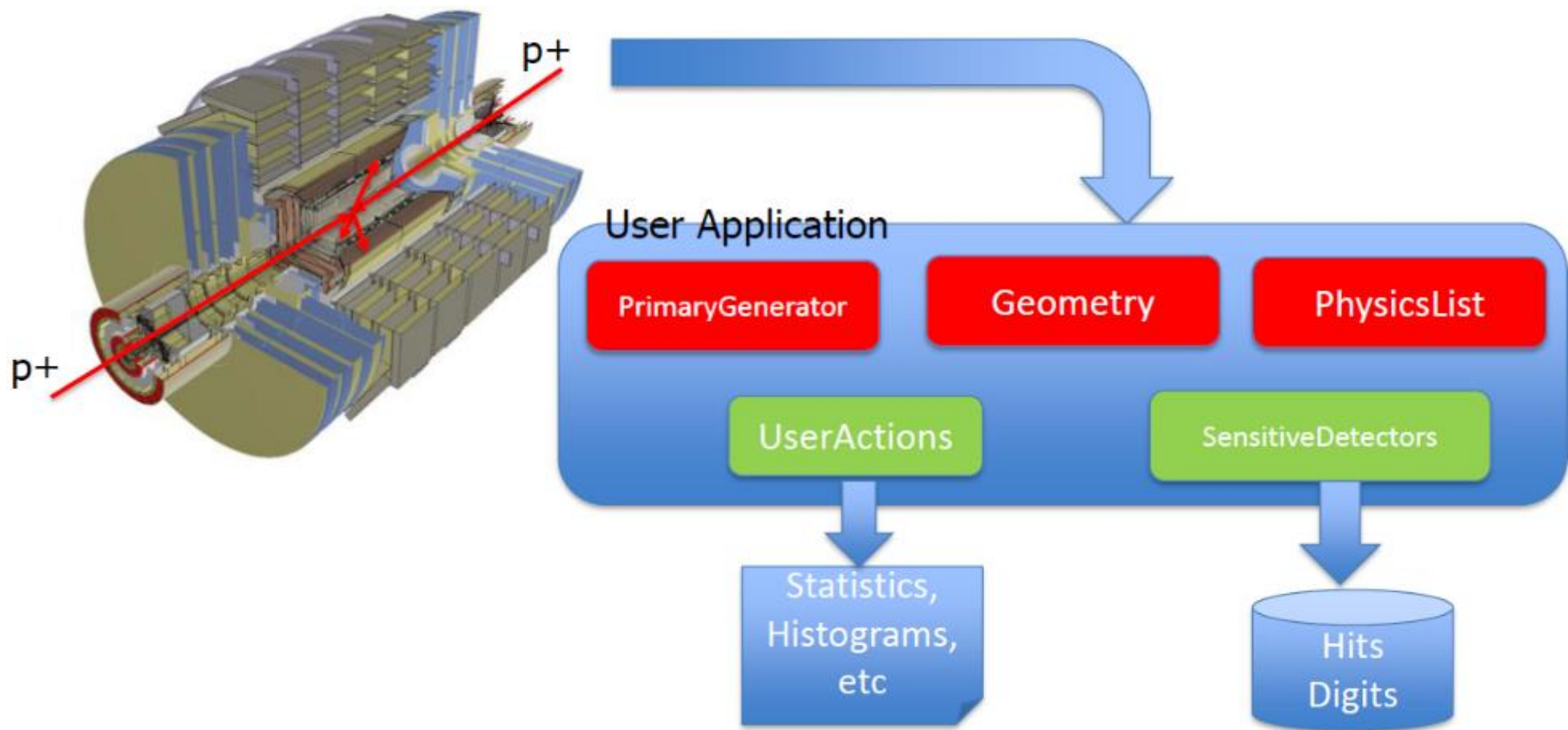
- wielowątkowość („multithreading”),
- własne interfejsy („messengers”),
- interfejs Roota (histogramy, n-tuple), interfejs python
- redukcja wariancji, „physics biasing”, „event biasing”, „geometrical biasing”,
- fotony optyczne, fizyka hadronowa, procesy i cząstki użytkownika,
- obciążenia energetyczne 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:

- M. Asai, „*Primary Particle*”, 4 – 8 February 2019, First Geant4 Tutorial Workshop in São Paulo
- V. Ivantchenko, „*Primary Generators*”, 25 – 31 May 2021, Geant4 Beginners Course @ CERN

Co jest potrzebne do uruchomienia symulacji?



- Użytkownik **musi** dostarczyć „źródło” cząstek pierwotnych aby Geant4 „wiedział” od czego ma zacząć symulację

Źródło – „Primary vertex” i „primary particle”

G4PrimaryVertex objects
= {position, time}



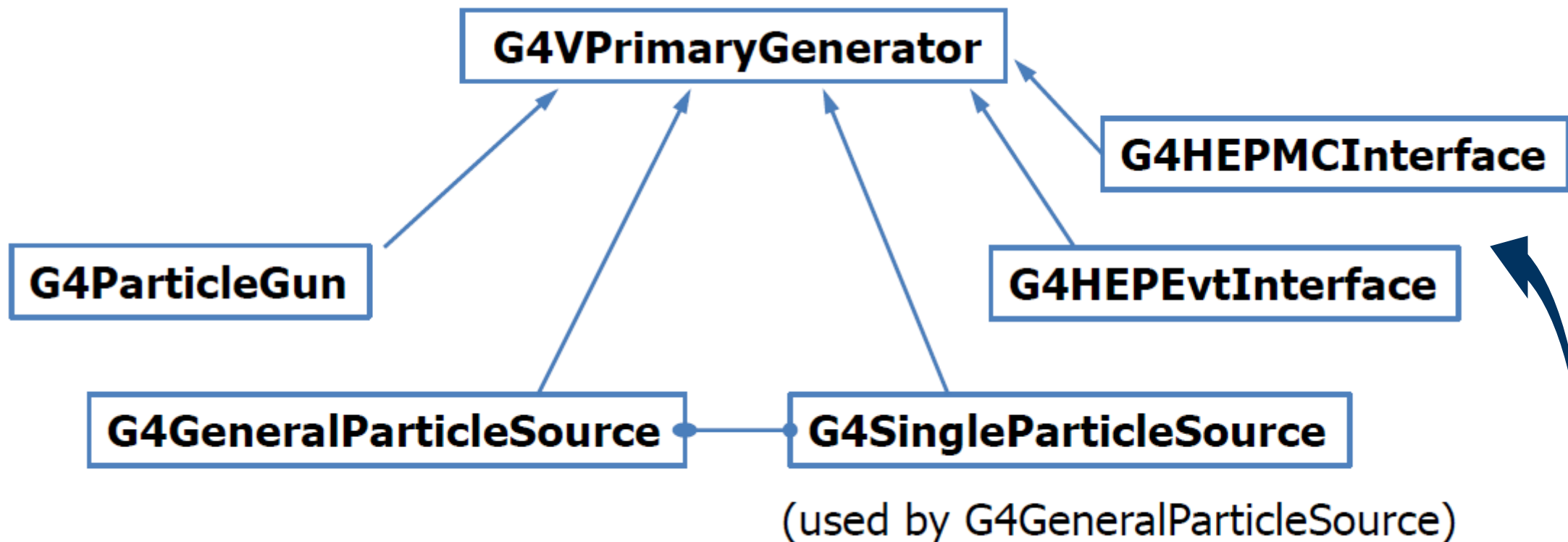
G4PrimaryParticle objects
= {PDG, momentum,
polarization...}

- Źródło cząstek pierwotnych musi na początku każdego eventu wygenerować co najmniej dwa obiekty, które razem opisują początkowy stan cząstki pierwotnej. Geant4 rozróżnia:
 - „**primary vertex**” – (wierzchołek) obiekt zawierający informację o początkowej pozycji i czasie pojawienia się cząstki pierwotnej (nie musimy startować w $t = 0$ s), oraz wskaźnik do obiektu reprezentującego cząstkę pierwotną
 - „**primary particle**” – obiekt zawierający informację o początkowym stanie cząstki pierwotnej, z którą rozpoczyna się event:
 - ID, wektor pędu, opcjonalnie wektor polaryzacji.
- Event może posiadać wiele „vertexów” jak i wiele cząstek pierwotnych.
- Z pojedynczym „vertexem” może być powiązanych więcej niż jedna cząstka pierwotna (vertex ma w sobie wektor wskaźników do cząstek pierwotnych)

Źródło - G4VUserPrimaryGeneratorAction

- Bazową klasą abstrakcyjną, która definiuje interfejs do sterowania źródłem jest **G4VUserPrimaryGeneratorAction**.
- Obiekt klasy pochodnej od **G4VUserPrimaryGeneratorAction** musi być obowiązkowo utworzony i zarejestrowany do RunManagera.
Powinno to być zrobione w metodzie **Build()** klasy **G4VUserActionInitialization**.
- **G4VUserPrimaryGeneratorAction** nie jest generatorem (źródłem) tylko interfejsem do generatora.
RunManager kontroluje generację cząstek pierwotnych za pośrednictwem **G4VUserPrimaryGeneratorAction**.
- **G4VUserPrimaryGeneratorAction** nie powinna sama generować cząstek pierwotnych ale utworzyć i wywoływać osobny generator cząstek pierwotnych (obiekt klasy pochodnej od **G4VPrimaryGenerator**).
- Geant4 zawiera kilka konkretnych, bardzo przydatnych implementacji źródeł np., **G4ParticleGun**, **G4GeneralParticleSource**.

- **G4VUserPrimaryGeneratorAction** nie jest generatorem tylko interfejsem do generatora. RunManager kontroluje generację cząstek pierwotnych za pośrednictwem **G4VUserPrimaryGeneratorAction**.
- **G4VUserPrimaryGeneratorAction** nie powinna sama generować cząstek pierwotnych ale utworzyć i wywoływać osobny generator cząstek pierwotnych (obiekt klasy pochodnej od **G4VPrimaryGenerator**).
- Konstruktor **G4VUserPrimaryGeneratorAction::G4VUserPrimaryGeneratorAction()**:
 - Utworzenie obiektu reprezentującego źródło
 - Konfiguracja początkowa źródła
- Metoda **G4VUserPrimaryGeneratorAction::GeneratePrimaries()**
 - Wywoływana na początku każdego eventu
 - Umożliwia randomizację ustawień generatora dla każdej cząstki z osobna
 - Wywołanie metody **GeneratePrimaryVertex()** generatrac(ów)



*Interfejsy do odczytu danych nt. cząstek
pierwotnych wytworzonych przez zewnętrzne
generatory.
Używane w fizyce wysokich energii.*

- Konkretna implementacja klasy G4VPrimaryGenerator.
- „Wystrzeliwuje” jedną (lub kilka identycznych) cząstkę pierwotną o zadanej energii, zadanego punktu, w zadanym kierunku i w zadanej chwili czasu.
- Udostępnia metody typu „set” do zadawania wartości początkowych.
- Początkowe wielkości można również wygodnie ustawiać za pomocą komend tekstowych z poziomu UI.
- W PrimaryGeneratorAction::GeneratePrimaries można wywoływać G4ParticleGun wielokrotnie z różnymi ustawieniami. W evencie pojawi się tyle cząstek pierwotnych ile razy zostało wywołane G4ParticleGun::GeneratePrimaryVertex(). Równolegle do G4ParticleGun można też wywoływać inne generatory – wszystkie wytworzone cząstki pierwotne pojawią się na początku eventu.

	G4ParticleGun ()
	G4ParticleGun (G4int numberofparticles)
	G4ParticleGun (G4ParticleDefinition *particleDef, G4int numberofparticles=1)
virtual	~G4ParticleGun ()
virtual void	GeneratePrimaryVertex (G4Event *evt)
void	SetParticleDefinition (G4ParticleDefinition *aParticleDefinition)
void	SetParticleEnergy (G4double aKineticEnergy)
void	SetParticleMomentum (G4double aMomentum)
void	SetParticleMomentum (G4ParticleMomentum aMomentum)
void	SetParticleMomentumDirection (G4ParticleMomentum aMomentumDirection)
void	SetParticleCharge (G4double aCharge)
void	SetParticlePolarization (G4ThreeVector aVal)
void	SetNumberOfParticles (G4int i)
G4ParticleDefinition *	GetParticleDefinition () const
G4ParticleMomentum	GetParticleMomentumDirection () const
G4double	GetParticleEnergy () const
G4double	GetParticleMomentum () const
G4double	GetParticleCharge () const
G4ThreeVector	GetParticlePolarization () const
G4int	GetNumberOfParticles () const

```

└─ gun
    List
    particle
    direction
    energy
    momentum
    momentumAmp
    position
    time
    polarization
    number
    ion
    ionL

```

Źródło – G4ParticleGun przykład użycia

Konstruktor.
Wywoływany
tylko raz.

Tworzymy obiekt
typu
G4ParticleGun.

Ustawiamy
domyślne
parametry źródła.

```
MyPrimaryGeneratorAction::MyPrimaryGeneratorAction()
```

```
{  
    G4int n_particle = 1;  
    fparticleGun = new G4ParticleGun(n_particle);  
    // default particle kinematic  
    G4ParticleTable* particleTable = G4ParticleTable::GetParticleTable();  
    G4ParticleDefinition* particle = particleTable->FindParticle("gamma");  
    fparticleGun->SetParticleDefinition(particle);  
    fparticleGun->SetParticleMomentumDirection(G4ThreeVector(0.,0.,1.));  
    fparticleGun->SetParticleEnergy(100.*MeV);  
    fparticleGun->SetParticlePosition(G4ThreeVector(0.,0.,-50*cm));  
}
```

Uwaga na marginesie.

To jest metoda odziedziczona po klasie bazowej G4VPrimaryGenerator.
Dlatego nie jest wymieniona w dokumentacji na poprzednim slajdzie.

Metoda
automatycznie
wywoływana na
początku każdego
eventu

```
void MyPrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
```

```
{  
    // Tu można jeszcze zmienić parametry cząstki pierwotnej,  
    // np. wylosować kierunek pędu, itp.  
    // fparticleGun->SetParticleMomentum(G4RandomDirection());  
    fparticleGun->GeneratePrimaryVertex(anEvent);  
}
```

Uwaga na marginesie.

GeneratePrimaryVertex tworzy zarówno obiekt typu „Primary Vertex” jak i „Primary Particle”.
Są one powiązane: Vertex zawiera wskaźnik do Particle.
Stworzona cząstka pierwotna dopisywana jest do bieżącego Eventu, którego przetwarzanie się rozpoczyna.

- Konkretna implementacja klasy [G4VPrimaryGenerator](#).
- Źródło znacznie bardziej złożone i uniwersalne niż G4ParticleGun
- Zalecana konfiguracja z poziomu komend tekstowych UI.
- Dokładny opis jest w nieco nieoczekiwanym miejscu: „Book For Application Developers” -> „Getting Started with Geant4 - Running a Simple Example” -> „[Geant4 General Particle Source](#)”
- Możliwości:
 - Rozkłady przestrzenne:
 - jednorodne: punktowy, z powierzchni 2D (płaszczyzn, okręgów, pierścieni, elips, kwadratów, prostokątów), z powierzchni lub objętości 3D (kuli, elipsoidy, walca, równoległościanu), z objętości fizycznej zdefiniowanej w UserDetectorConstruction.
 - niejednorodny z elipsy wg. Gaussa (2D) – do symulacji wiązki akceleratorowej.
 - Rozkłady kątowe: jednokierunkowy, izotropowy, cosinusowy, Gaussa, zogniskowany lub dowolny (zdefiniowany przez użytkownika).
 - Widmo energetyczne: mono, liniowe, wykładnicze, potęgowe, Gaussowskie, ciała doskonale czarnego, interpolowane z danych.
 - Można jednocześnie zdefiniować wiele źródeł o różnych rozkładach i przypisać im odpowiednie współczynniki wagowe.

Wbudowane rozkłady energii

Spectrum	Abbreviation	Functional Form	User Parameters
mono-energetic	Mono	$I \propto \delta(E - E_0)$	Energy E_0
linear	Lin	$I \propto I_0 + m \times E$	Intercept I_0 , slope m
exponential	Exp	$I \propto \exp(-E/E_0)$	Energy scale-height E_0
power-law	Pow	$I \propto E^\alpha$	Spectral index α
Gaussian	Gauss	$I = (2\pi\sigma)^{-\frac{1}{2}} \exp[-(E/E_0)^2/\sigma^2]$	Mean energy E_0 , standard deviation σ
bremsstrahlung	Brem	$I = \int 2E^2 [h^2 c^2 (\exp(-E/kT) - 1)]^{-1}$	Temperature T
black body	Bbody	$I \propto (kT)^{\frac{1}{2}} E \exp(-E/kT)$	Temperature T (see text)
cosmic diffuse gamma ray	Cdg	$I \propto [(E/E_b)^{\alpha_1} + (E/E_b)^{\alpha_2}]^{-1}$	Energy range E_{\min} to E_{\max} ; energy E_b and indices α_1 and α_2 are fixed (see text)

Źródło - [G4GeneralParticleSource](#) – jak używać?

Źródło tworzymy jak zwykle, operatorem new, wewnątrz konstruktora klasy użytkownika PrimaryGeneratorAction().

W metodzie PrimaryGeneratorAction::GeneratePrimaries nie robimy nic poza wywołaniem metody GeneratePrimaryVertex na rzecz obiektu reprezentującego nasze źródło.

Konfigurację źródła zaleca się wykonać z poziomu interfejsu UI (lub makra) przed rozpoczęciem runu.

```
#include "PrimaryGeneratorAction.hh"
#include "G4GeneralParticleSource.hh"

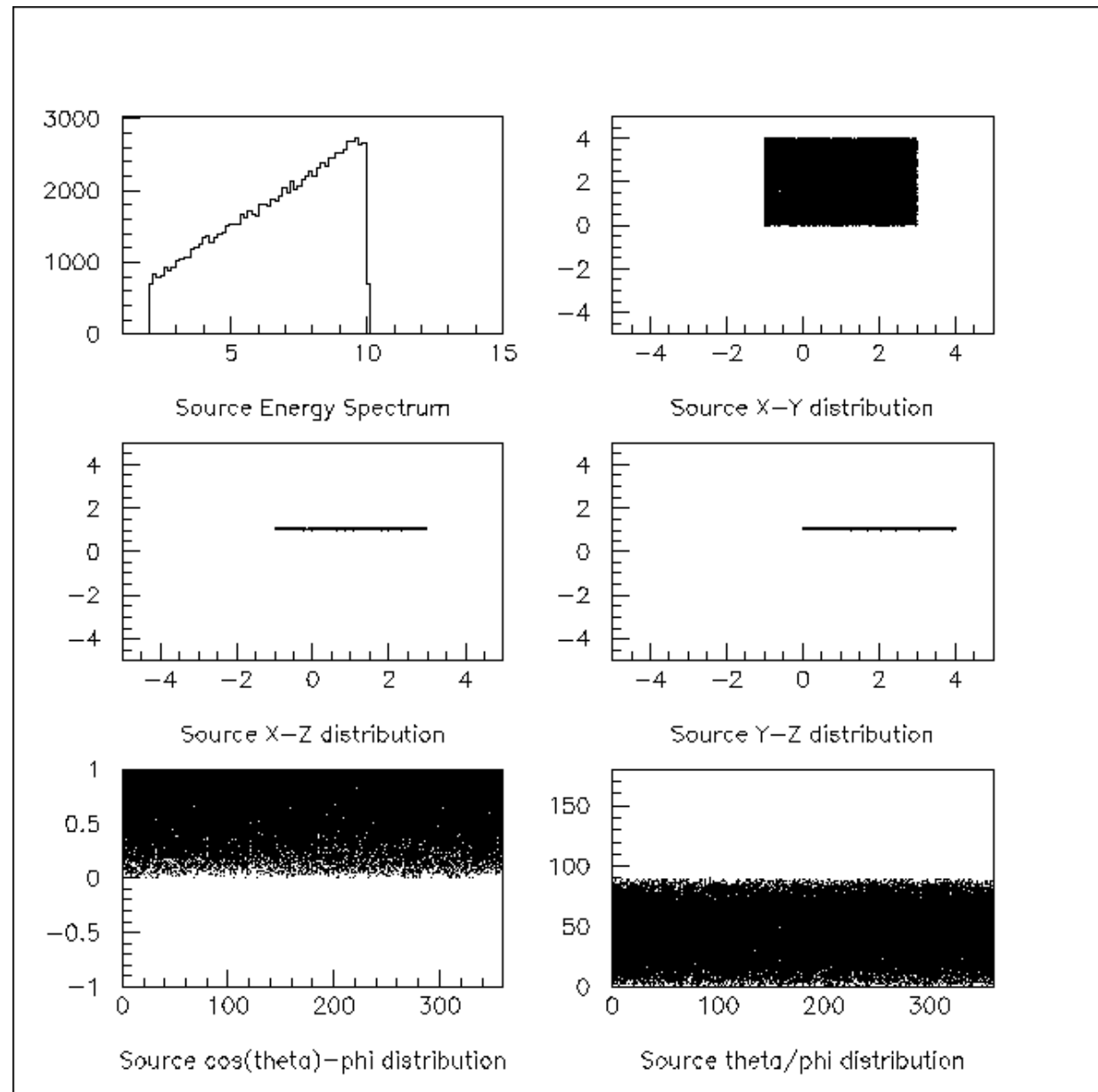
PrimaryGeneratorAction::PrimaryGeneratorAction()
: G4VUserPrimaryGeneratorAction(),
  fGenerator(0)
{
    fGenerator = new G4GeneralParticleSource;
}

void PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    fGenerator->GeneratePrimaryVertex(anEvent);
}
```

Źródło - G4GeneralParticleSource – jak używać?

```
# Macro test2.g4mac
/control/verbose 0
/tracking/verbose 0
/event/verbose 0
/gps/verbose 2
/gps/particle gamma
/gps/pos/type Plane
/gps/pos/shape Square
/gps/pos/centre 1 2 1 cm
/gps/pos/halfx 2 cm
/gps/pos/halfy 2 cm
/gps/ang/type cos
/gps/ene/type Lin
/gps/ene/min 2 MeV
/gps/ene/max 10 MeV
/gps/ene/gradient 1
/gps/ene/intercept 1
/run/beamOn 10000
```

← Domyślnie w płaszczyźnie prostopadłej do osi Z



<https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/html/GettingStarted/generalParticleSource.html#example-macro-file>

Źródło - [G4GeneralParticleSource](#) – jak używać?

Import widma energii z pliku.

- Rodzaj rozkładu energii definiujemy komendą `/gps/ene/type`.
Pozwala ona, m.in., zdefiniować typ **User** („user-defined histogram”) oraz typ **Arb** („point-wise spectrum”).
- Do wczytywania histogramu z pliku służy komenda `/gps/hist/file`.

UWAGA! DZIAŁANIE WBREW INTUICJI! Z pliku da się zaimportować widmo energii tylko dla typu **Arb**, a nie da się dla typu **User** (wbrew sugestii nt. „user-defined histogram”). W tym drugim przypadku trzeba zadać histogram kanał po kanale używając wielokrotnie polecenia `/gps/hist/point`.

Działający przykład:

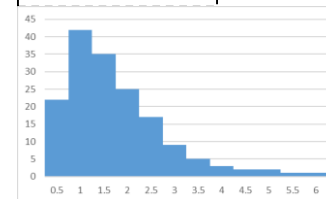
```
/gps/ene/type Arb          # Arb (point-wise spectrum)
/gps/hist/type arb         # Typ histogramu
/gps/hist/file widmo.dat   # Import an arbitrary energy histogram in an ASCII file.
                           # Uwaga! W dokumentacji jest to niezbyt wyraźnie powiedziane,
                           #       ale to działa tylko z opcją /gps/ene/type arb.
                           # The format should be one Ehi Weight pair per line of the file.
                           # For histograms, Ehi is the bin upper edge, in Geant4 default units
                           # (MeV for energy, rad for angle).

/gps/hist/inter Lin        # Kluczowa komenda. Bez tego się wysypuje!!!
                           # Sets the interpolation type (Lin linear, Log logarithmic, Exp
                           # exponential, Spline cubic spline) for point-wise spectra.
                           # This command must be issued immediately after the last data point.
```

```
# Currently histograms are limited to 1024 bins.
# The first value of each user input data pair is treated as the upper edge of the histogram
bin and the second value is the bin content. The exception is the very first data pair the user
input whose first value is the treated as the lower edge of the first bin of the histogram, and
the second value is not used. This rule applies to all distribution histograms, as well as
histograms for biasing.
```

widmo.dat

```
0.0 0.0
0.5 22.0
1.0 42.0
1.5 35.0
2.0 25.0
2.5 17.0
3.0 9.0
3.5 5.0
4.0 3.0
4.5 2.0
5.0 2.0
5.5 1.0
6.0 1.0
```

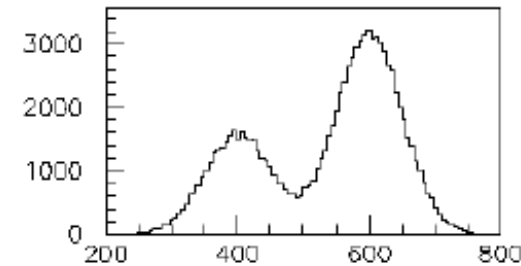


Źródło - G4GeneralParticleSource – jak używać?

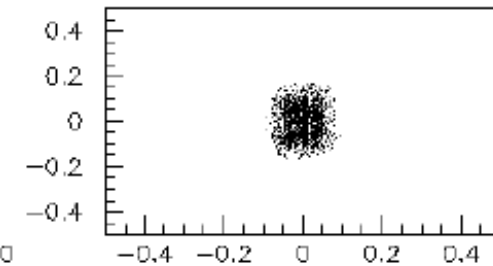
Przykład źródła składającego się z wiązki protonowej i elektronowej o różnych natężeniach

```
# beam #1
# default intensity is 1
# now change to 5.
/gps/source/intensity 5.
# this is a proton beam
/gps/particle proton
/gps/pos/type Beam
# the incident surface
# is in the y-z plane
/gps/pos/rot1 0 1 0
/gps/pos/rot2 0 0 1
# the beam spot is centered
# at the origin and is of
# 1d gaussian shape with
# a 1 mm central plateau
/gps/pos/shape Circle
/gps/pos/centre 0. 0. 0. mm
/gps/pos/radius 1. mm
/gps/pos/sigma_r .2 mm
# the beam is travelling along
# the X_axis with
# 5 degrees dispersion
/gps/ang/rot1 0 0 1
/gps/ang/rot2 0 1 0
/gps/ang/type beam1d
/gps/ang/sigma_r 5. deg
# the beam energy is in
# gaussian profile
# centered at 400 MeV
/gps/ene/type Gauss
/gps/ene/mono 400 MeV
/gps/ene/sigma 50. MeV
```

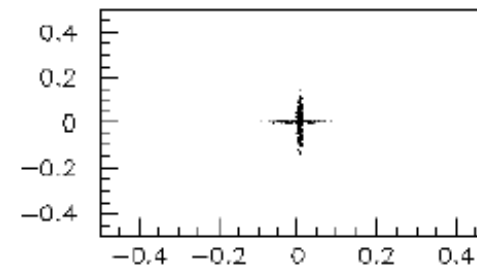
```
# beam #2
# 2x the intensity of beam #1
/gps/source/add 10.
# this is a electron beam
/gps/particle e-
/gps/pos/type Beam
# it beam spot is of 2d
# gaussian profile
# with a 1x2 mm2 central
# plateau
# it is in the x-y plane
# centred at the origin
/gps/pos/centre 0. 0. 0. mm
/gps/pos/halfx 0.5 mm
/gps/pos/halfy 1. mm
/gps/pos/sigma_x 0.1 mm
# the spread in y direction
# is stronger
/gps/pos/sigma_y 0.2 mm
# the beam is travelling
# along -Z_axis
/gps/ang/type beam2d
/gps/ang/sigma_x 2. deg
/gps/ang/sigma_y 1. deg
# gaussian energy profile
/gps/ene/type Gauss
/gps/ene/mono 600 MeV
/gps/ene/sigma 50. MeV
```



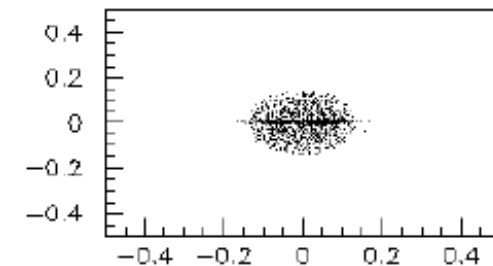
Source Energy Spectrum



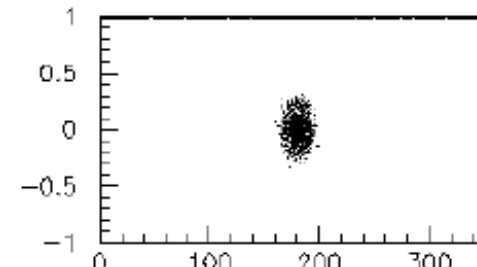
Source X-Y distribution



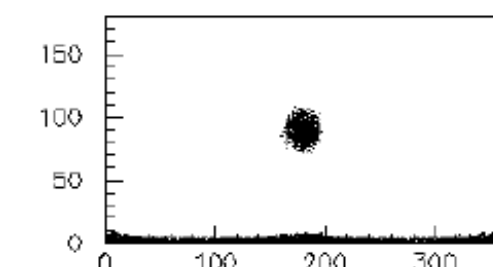
Source X-Z distribution



Source Y-Z distribution



Source cos(theta)-phi distribution



Source theta/phi distribution

Particle Gun

- Proste, do podstawowych zastosowań.
- Łatwe w obsłudze (tylko kilka komend UI).
- Wbudowane metody do konfiguracji z poziomu kodu (track-by-track lub event-by-event).
- Generuje jedną cząstkę z punktu.

General Particle Source (GPS)

- Znacznie większe możliwości.
- Obsługa z poziomu UI (z poziomu kodu możliwa ale bardziej uciążliwa).
- Potrafi wystrzeliwać cząstki z powierzchni lub objętości brył.
- Potrafi losować energię, kąt, pozycję według zadanych rozkładów.
- Potrafi importować rozkłady z plików.
- Pozwala zaimplementować wiele różnych źródeł jednocześnie.

Kiedy, które źródło. Zalecenia.

- GPS jeśli musisz wystrzeliwać cząstki z powierzchni lub objętości (na zewnątrz lub do wewnątrz).
- GPS jeśli potrzebujesz rozkładów energii/kąta bardziej złożonych niż Gauss.
- W przeciwnych wypadkach użyj Particle Gun.

Subiektywna opinia: *zawsze używam GPS ze względu na elastyczność i możliwość ewentualnej rozbudowy źródła bez zmiany kodu.*

Dziękuję za uwagę



NARODOWE
CENTRUM
BADAŃ
JĄDROWYCH
ŚWIERK

www.ncbj.gov.pl