
ĆWICZENIE 60

ANALIZA DANYCH PRZY UŻYCIU PAKIETU ROOT

II PRACOWNIA FIZYCZNA
UNIwersytet Śląski w Katowicach

Przed rozpoczęciem ćwiczenia należy zapoznać się dokładnie z instrukcją!
Ćwiczenie przeznaczone jest dla studentów posiadających podstawowe umiejętności programowania w C++.

1 CEL ĆWICZENIA

Celem ćwiczenia jest zapoznanie się z możliwościami pakietu ROOT oraz przeprowadzenie przy jego użyciu analizy danych pomiarowych pochodzących z jednego z doświadczeń dostępnych na Pracowni Jądrowej w Instytucie Fizyki U. Śl.

2 WSTĘP

2.1 Co to jest ROOT?

Środowisko ROOT oparte na popularnym języku programowania C++, stworzone zostało do analizy wyników otrzymywanych w eksperymentach fizyki jądra i cząstek elementarnych. Powstało w Europejskim Ośrodku Badań Jądrowych CERN pod Genewą w 1994 roku na potrzeby tamtejszych eksperymentów akceleratorowych z dziedziny fizyki wysokich energii, na początku głównie dla NA49. W jego skład wchodzi między innymi: interpreter i kompilator C++, szereg przydatnych bibliotek, w tym biblioteka matematyczna, biblioteka statystyczna, biblioteka graficzna, biblioteka do tworzenia histogramów oraz narzędzia do równoległej analizy danych. Dzięki zastosowaniu klas obudowujących funkcje systemu operacyjnego, aplikacje napisane w środowisku ROOT mogą być uruchamiane pod wieloma systemami, np. pod Linux'em czy Windows'em. Oprócz jego oczywistych zalet jakimi są prostota, kompatybilność i szerokie zastosowanie w fizyce jądra i cząstek elementarnych, niewątpliwie jedną z ważniejszych jest także wolność dostępu. Środowisko root można bezpłatnie ściągnąć ze strony internetowej [1]

2.2 Moduły ROOT'a.

- BINDINGS - powiązanie (binding) generalnie odnosi się do przyporządkowywania (mapowania) jednych zasobów systemowych do drugih. W kontekście bibliotek softwareowych, bindings to biblioteki, które łączą 2 języki programowe w taki sposób, aby biblioteka napisana w jednym języku mogła być używana w innym. ROOT posiada powiązania do Pythona i Ruby.
- CINT - Command Interpreter (CINT) to interpreter C/C++.
- CORE - moduł ten dostarcza niezbędnych funkcji używanych przez ROOT'a.
- GEOM - moduł umożliwia używanie, tworzenie a także wyświetlanie geometrii detektorów.
- GRAPH2D, GRAPH3D - umożliwia tworzenie oraz szereg operacji na 2 i 3 wymiarowym grafie, będących zbiorem współrzędnych x, y (2D) oraz z (3D).
- GUI - Graphic User Interface (GUI) dostarcza elementów umożliwiających budowę aplikacji "okienkowych", takich jak przeglądarka (TBrowser) czy wizualizator geometrii.
- HIST - umożliwia tworzenie histogramów 2D oraz 3D oraz szereg operacji na nich, w tym statystycznych.
- HTML - dostarcza funkcji umożliwiających automatyczną dokumentację klas w formacie html.
- IO - umożliwia szereg operacji wejścia/wyjścia, takich jak odczyt i zapis do pliku, komunikacja z bazami danych SQL a także z CERN'owskim magazynem danych (CASTOR).
- MATH - bardzo ważny moduł, umożliwiający przeprowadzanie operacji matematycznych, począwszy od obliczania podstawowych zależności, skończywszy na algorytmach regresji, operacjach na macierzach czy wektorach.
- MISC - różne funkcje nie podпадаjące pod inne kategorie.
- MONTECARLO - zbiór funkcji umożliwiających przeprowadzanie symulacji zjawisk fizycznych za pomocą techniki Monte Carlo.
- NET - moduł ten jest odpowiedzialny za komunikację frameworku ROOT za pośrednictwem sieci.
- PROOF - moduł zawierający algorytmy umożliwiające przetwarzanie równoległe.
- ROOFIT - jest to biblioteka dostarczająca narzędzi do matematycznego modelowania zdarzeń fizycznych.
- SQL - moduł służący do przetwarzania zapytań SQL.
- TMVA - zbiór klas służących do analizy danych wielu zmiennych.

- TREE - dostarcza kontenerów opartych na drzewiastych strukturach danych.

3 Zagadnienia teoretyczne

Wiadomości na temat histogramu, grafu, dopasowywania funkcji do danych oraz funkcji klasy TMath [2], [3]. Przykładowe pytania do kolokwium:

- Co to jest histogram? Podaj przykłady.
- Jak należy rozumieć "bin" histogramu? Podaj przykłady.
- Jaki histogram otrzymamy w wyniku operacji: `h1.Scale(1.0/h1.GetEntries())`?
- Czym różnią się komendy: `hist->Fill(liczba)` oraz `hist->SetBinContent(bin, liczba)`? W jakich sytuacjach korzystamy z pierwszej, a w jakich z drugiej metody wypełniania histogramu?
- Podaj przykłady operacji na histogramach dostępnych w pakiecie ROOT.
- Jakie znasz opcje rysowania histogramu w ROOT?
- Co to jest graf w ROOT?
- Omów funkcję Gaussa.
- Omów dopasowanie metodą regresji liniowej.
- Jakie są predefiniowane funkcje matematyczne w pakiecie ROOT?
- Czy w pakiecie ROOT użytkownik może zdefiniować własną funkcję? Podaj przykład.
- Czy w pakiecie ROOT można dopasować funkcję do histogramu w ograniczonym zakresie? W jaki sposób?

4 Przebieg ćwiczenia

4.1 Ćwiczenia wprowadzające

Jak zostało powiedziane wcześniej, CINT to interpreter C/C++. Aby uruchmić ROOT'a, należy w linii komend systemu operacyjnego wpisać polecenie: `root`. Uruchomienie skryptu następuje po wpisaniu komendy `.x nazwa_skryptu.C`. Wyjście z ROOT'a następuje po wpisaniu polecenia `.q`

Utwórz swój podkatalog (nazwa katalogu - nazwisko studenta) w katalogu Dane Pracownia. Ze strony [4] ściągnij pliki z danymi.

4.1.1 Histogramy i Grafy

W swoim katalogu stwórz plik `MojPierwszyHistogram.C` oraz `MojPierwszyGraf.C` i wpisz do nich odpowiedni kod:

```

1  void MojPierwszyHistogram()
2  {
3      TH1F *h1 = new TH1F("h1", "Moj pierwszy histogram",100,-3,3);
4      TCanvas *c1 = new TCanvas("c1","c1");
5      FILE *fp = fopen("dane_hist.dat","r");
6      Int_t IloscZliczen;
7      Int_t bin=1;
8      while(!feof(fp)) {
9          fscanf(fp,"%d",&IloscZliczen);
10         h1->SetBinContent(bin,IloscZliczen);
11         bin++;
12     }
13     fclose(fp);
14     h1->GetXaxis()->SetTitle("Wartosc");
15     h1->GetYaxis()->SetTitle("Ilosc Zliczen");
16     h1->Draw("E");
17     h1->Fit("gaus");
18     c1->SaveAs("MojPierwszyHistogram.pdf");
19 }

1  void MojPierwszyGraf()
2  {
3      TCanvas *c1 = new TCanvas("c1","c1");
4      FILE *fp = fopen("dane_graf.dat","r");
5      Int_t x[100], y[100];
6      Int_t licz=0;
7      while(!feof(fp)) {
8          fscanf(fp,"%d %d",&x[licz], &y[licz]);
9          cout<<x[licz]<<"\t"<<y[licz]<<endl;
10         licz++;
11     }
12     fclose(fp);
13
14     TGraph *g1 = new TGraph(100,x,y);
15     g1->GetXaxis()->SetTitle("X");
16     g1->GetYaxis()->SetTitle("Y");
17     g1->SetMarkerStyle(20);
18     g1->Draw("ap");
19     g1->Fit("pol1");
20     //TF1 *MojaFunkcja = new TF1("MojaFunkcja", "pol1", 20,50);
21     //g1->Fit("MojaFunkcja","R");
22     c1->SaveAs("MojPierwszyGraf.pdf");
23 }

```

Uruchom skryptu z linii komend ROOT'a. Omów poszczególne linie każdego skryptu.

4.1.2 Błędy zawartości binu w histogramie

W skrypcie `MojPierwszyHistogram.C` wywołano funkcję `Draw` wraz z opcją `"E"`. Spowodowało to narysowanie histogramu wraz z błędem zawartości binu, który standardowo obliczany jest jako \sqrt{N} . Jednakże w przypadku wypełniania histogramu wagami (`hist->Fill(bin, w)`), bądź też w sytuacji, gdy przeprowadzane są operacje arytmetyczne na histogramach, przed wypełnieniem histogramu należy wywołać funkcję `Sumw2()`:

```
hist->Sumw2();
```

Wówczas błąd obliczany będzie jako pierwiastek sumy wag (zgodnie z regułą przenoszenia błędu wielkości nieskorelowanych): $\sqrt{\sum w_i^2}$.

Zastanów się jaki będzie błąd zawartości binu w histogramie, gdy została wywołana funkcja `Sumw2()` i `w=1`

Błąd można ustawić również samodzielnie komendą: `hist->SetBinError(bin, error);`

4.1.3 Działania na histogramach

Histogramy można dodać, odjąć, pomnożyć bądź podzielić. Operacji tych można dokonywać na histogramach o tej samej ilości binów. Stwórz plik `OdejmowanieHistogramow.C` i wpisz do niego następujący kod:

```
1  void OdejmowanieHistogramow()
2  {
3      TH1F *hprobka = new TH1F("hproba", "Widmo Promieniowania X",
4                               ,8000, 0, 8000);
5      TH1F *htlo = new TH1F("htlo", "Widmo Tla",8000, 0, 8000);
6      TH1F *hroznica = new TH1F("hroznica", "Widmo Po Odjeciu Tla",
7                               ,8000, 0, 8000);
8
9      TCanvas *c1 = new TCanvas("c1","c1");
10     c1->Divide(3,1);
11
12     FILE *fp = fopen("PromXProbka.dat","r");
13     Int_t IloscZliczen;
14     Int_t bin=1;
15     while(!feof(fp)) {
16         fscanf(fp,"%d",&IloscZliczen);
17         hprobka->SetBinContent(bin,IloscZliczen);
18         bin++;
19     }
20     fclose(fp);
21
22     FILE *fp = fopen("Tlo.dat","r");
23     IloscZliczen = 0;
24     bin=1;
25     while(!feof(fp)) {
26         fscanf(fp,"%d",&IloscZliczen);
27         htlo->SetBinContent(bin,IloscZliczen);
28         bin++;
29     }
30 }
```

```

27     fclose(fp);
28
29     c1->cd(1);
30     hprobka->Draw("E");
31     hprobka->GetXaxis()->SetTitle("Wartosc");
32     hprobka->GetYaxis()->SetTitle("Ilosc Zliczen");
33     hprobka->GetYaxis()->SetRangeUser(0,400);
34     Double_t bladprobka = hprobka->GetBinError(3950);
35     cout<<hprobka->GetBinContent(3950)<<"\t"
36         <<bladprobka<<endl;
37
38     c1->cd(2);
39     htlo->Draw("E");
40     htlo->GetXaxis()->SetTitle("Wartosc");
41     htlo->GetYaxis()->SetTitle("Ilosc Zliczen");
42     htlo->GetYaxis()->SetRangeUser(0,400);
43     Double_t bladtlo = htlo->GetBinError(3950);
44     cout<<htlo->GetBinContent(3950)<<"\t"
45         <<bladtlo<<endl;
46
47     Double_t blad1 = TMath::Sqrt(bladprobka*bladprobka+bladtlo*
        bladtlo);
48     c1->cd(3);
49     hroznica->Sumw2();
50     hroznica->Add(hprobka, htlo, 1, -1);
51     hroznica->Draw("E");
52     Double_t blad2 = hroznica->GetBinError(3950);
53     cout<<hroznica->GetBinContent(3950)<<"\t"<<
54         blad1<<"\t"<<
55         blad2<<endl;
56
57 }

```

Uruchom skrypt z linii komend ROOT'a i omów poszczególne linie kodu.

Uruchom skrypt dwukrotnie, raz z zakomentowaną i raz z odkomentowaną linią nr 49. Omów różnice w wartościach błędów wypisywanych na ekran w jednym i drugim przypadku.

4.2 Opracowanie wyników z wykorzystaniem pakietu ROOT.

Prowadzący laboratorium dostarczy danych pomiarowych pochodzących z jednego z doświadczeń dostępnych na Pracowni Jądrowej w Instytucie Fizyki U. Śl. Postępując zgodnie z instrukcją do tego ćwiczenia należy w trakcie zajęć przeprowadzić analizę danych wykorzystując pakiet ROOT oraz sporządzić sprawozdanie. Do sprawozdania należy dołączyć swój skrypt.

Jeżeli nie jesteś pewna/pewny czy w danym przypadku należy wywołać funkcję Sumw2(), wywołaj ją zawsze.

Dlaczego nie jest to błędem?

References

- [1] root.cern.ch
- [2] <http://root.cern.ch/drupal/content/root-users-guide-534>, Rozdziały 3, 4, 5
- [3] <http://root.cern.ch/root/html534/TMath.html>
- [4] <http://www.nuph.us.edu.pl/~kschmidt/PracowniaJadrowa/>